Indhold

Asymptotisk notation

Opgave 1 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$n \cdot \log n$ er $O(1)$		
$n^{0.001}$ er $O(\log n)$		
$n \cdot \log n$ er $O(\log n)$		
$n \cdot \log n$ er $O(5^5)$		
$\log n + 5n \cdot \log n$ er $O(n^3)$		
$\log n$ er $O(n^2)$		
$n^{0.1}$ er $O(\log n)$		
$6n^3$ er $O(\sqrt{n})$		
$(\log n)^3$ er $O(8^{\log n})$		
$n^{2/3} \cdot n^{1/3}$ er $\Theta(n^2)$		
$7n \cdot \log n$ er $\Omega(\log n)$		
n er $\Theta(n^{2/3} \cdot n^{1/3})$		

Opgave 2 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$n^2 \log n$ er $O(3^3)$		
$\log n^2$ er $O(1)$		
$6\sqrt{n}$ er $O(n\sqrt{n})$		
$\sqrt{n} + \sqrt{n}$ er $O(n \cdot \log n)$		
$n \cdot \log n$ er $O((\log n)^3)$		
2^n er $O(\sqrt{n} \cdot \log n)$		
$n\sqrt{n}$ er $O(n^{3/2})$		
$8^{\log n}$ er $O(n^{2/3})$		
$n \cdot \log n$ er $O((\log n)^2)$		
3^n er $\Omega(n^n)$		
$7n \cdot \log n$ er $\Theta(\log(n!))$		
$\sum_{i=1}^{n} i$ er $\Omega(\sqrt{n} \cdot \log n)$		

Opgave 3 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$2^{\log n} + 5n^n$ er $O(\sqrt{n} \cdot \log n)$		
$n^2 \log n$ er $O(\sqrt{n})$		
$\sqrt{n} \cdot \log n$ er $O(n \cdot \log n)$		
$4n^2$ er $O(n^{0.1})$		
n^2 er $O(\log n^2)$		
2^n er $O(n \cdot \log n)$		
n er $O(\log n)$		
$2^{3 \log n}$ er $O(n \cdot \log n)$		
5 er $O(n \cdot \log n)$		
$n^{0.01}$ er $\Theta(\sqrt{n})$		
$\sqrt{n}$ er $\Omega(\log n^2)$		
$2^{\log n}$ er $\Omega(n^2)$		

Opgave 4 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$(\log n)^3 + 2^{3 \log n}$ er $O(8^{\log n})$		
$(\log n)^2$ er $O(\sqrt{n})$		
n^2 er $O(n)$		
$\sqrt{n}$ er $O(n \cdot \log n)$		
$5 \cdot 2^{3 \log n}$ er $O(\sqrt{n})$		
$(\log n)^6$ er $O(n^2)$		
$\sqrt{n} \cdot \log n + \sqrt{n}$ er $O(1)$		
$2 \log(n!)$ er $O(n)$		
$2^{2 \log n}$ er $O(n^{0.001})$		
n^2 er $\Omega(2^n)$		
$4n^{1/3}$ er $\Theta(n^{1/3})$		
$n^{0.1}$ er $\Theta(\sqrt{n} \cdot \log n)$		

Opgave 5 (6 %)

check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$(\log n)^7/5$ er $O(n^{0.01})$		
$4^{\log n}$ er $O(\log(n!))$		
$(\log n)/7$ er $O(\sqrt{n})$		
$n^{0.01}$ er $O(2^{3\log n})$		
$\sqrt{n}\cdot\log n$ er $O(n^2)$		
$7\cdot 4^{\log n}$ er $O((\log n)^3)$		
n^2 er $O(n^{1/3})$		
$(\log n)^3 + n^2/6$ er $O(\log(n!))$		
$2^{\log n} + (\log n)^3$ er $O(\sqrt{n})$		
3^n er $\Omega(n^3)$		
$n\cdot\log n + \log n$ er $\Theta(n\cdot\log n)$		
$\log n^2$ er $\Theta(\log n)$		

Opgave 6 (6 %)

check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$\log n$ er $O(n)$		
$3n^{0.001}$ er $O(\sqrt{n})$		
$5n^2$ er $O(n!)$		
$2^{3\log n}/5$ er $O(\log n)$		
n^2 er $O(3^n)$		
$n^{0.1}$ er $O(n!)$		
$3\cdot 2^{3\log n}$ er $O(\sqrt{n})$		
$n + n/6$ er $O(n^2)$		
$\sqrt{n}\cdot\log n$ er $O(\log n)$		
$n^{0.001} + \log n^2$ er $\Omega(n^2\log n)$		
$4^{\log n}$ er $\Omega(3^n)$		
$2/7$ er $\Omega(n^{0.001})$		

Opgave 7 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$n^{0.001}/4$ er $O(8^{\log n})$		
$\log(n!)/6$ er $O(2^{\log n})$		
$\sqrt{n}$ er $O(\log n)$		
$n^2 \log n$ er $O(n^{2/3})$		
$\log n$ er $O((\log n)^4)$		
$2^{3 \log n}$ er $O((\log n)^7)$		
$(\log n)^2$ er $O(n^{2/3})$		
$(\log n)^3/2$ er $O(8^{\log n})$		
$n^{2/3}$ er $O((\log n)^3)$		
$n \cdot \log n$ er $\Omega(\log n)$		
3^3 er $\Omega(n^{3/2})$		
$n^{3/2}$ er $\Omega(n^{0.1})$		

Opgave 8 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$\log n^2$ er $O(\sqrt{n})$		
2 er $O((\log n)^2)$		
$n^{1/3}$ er $O(\sqrt{n})$		
$5^5 + 3 \cdot 2^2$ er $O(\sum_{i=1}^{n} i)$		
$n + n \cdot \log n$ er $O(n^2)$		
$5\sqrt{n} \cdot \log n$ er $O(n^2)$		
n er $O(\sqrt{n})$		
$(\log n)^3$ er $O(\sqrt{n})$		
$2\sqrt{n}$ er $O(3^3)$		
$\log(n!)$ er $\Omega(\sqrt{n})$		
$6n \cdot \log n$ er $\Omega(n^{0.001})$		
$n!$ er $\Omega(\sqrt{n})$		

Opgave 9 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
$\sqrt{n} \cdot (\log n)/6$ er $O(n \cdot \log n)$		
$4(\log n)^3$ er $O(n^{0.001})$		
$\sqrt{n}$ er $O(n^{0.01})$		
$\log n$ er $O(n^2)$		
$\log n + \log(n!)$ er $O(n^2 \log n)$		
$7 \cdot 3^n + \sqrt{n} \cdot \log n$ er $O(n \cdot \log n)$		
$\log(n!) + n^{2/3}$ er $O(n \cdot \log n)$		
$\log n + 5^5$ er $O(n \cdot \log n)$		
$4n^2 + (\log n)^5$ er $O(n)$		
$(\log n)^6/3$ er $\Theta(n^{0.1})$		
n^2 er $\Omega(n^2 \log n)$		
$n^{3/2}$ er $\Theta(\sum_{i=1}^{n} i)$		

Opgave 10 (6 %) check

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
n^2 er $O(\sqrt{n})$		
$4 \log n$ er $O(n \cdot \log n)$		
$\log n$ er $O(n \cdot \log n)$		
$n^{1/3}$ er $O(n^2 \log n)$		
$2 \log n$ er $O(n^{2/3})$		
$n \cdot \log n$ er $O(n\sqrt{n})$		
$3n^n$ er $O(n^2 \log n)$		
$n^{2/3}$ er $O(n^2 \log n)$		
$1 + 3$ er $O((\log n)^3)$		
$\log n + n \cdot \log n$ er $\Theta(n \cdot \log n)$		
n^3 er $\Theta(n\sqrt{n})$		
$n \cdot \log n$ er $\Theta(n)$		

Analyse af løkker

Opgave 11 (6 %) check

Algoritme loop1(n)
  $s = 0$
  for $i = 1$ **to** n
    for $j = 1$ **to** $i * i$
      $s = s + 1$

Algoritme loop2(n)
  $s = 1$
  for $i = 1$ **to** n
    for $j = 1$ **to** n
      $s = s + 1$

Algoritme loop3(n)
  $i = 0$
  $s = 0$
  while $s \leq n$
    $i = i + 1$
    $s = s + i$

Algoritme loop4(n)
  for $i = 0$ **to** n
    $j = 0$
    $s = 0$
    while $s \leq i$
      $j = j + 1$
      $s = s + j$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n^3)$	$\Theta((\log n)^2)$	$\Theta(\sqrt{n})$	$\Theta(n^2)$	$\Theta(n \log n)$	$\Theta(n\sqrt{n})$	$\Theta(\log n)$	$\Theta(n)$
loop1								
loop2								
loop3								
loop4								

Opgave 12 (6 %) check

```
Algoritme loop1(n)
  s = 1
  for i = n to 1 step −1
    s = s + 1
```

```
Algoritme loop2(n)
  for i = 1 to n
    j = i
    while j > 0
      j = j − 1
```

```
Algoritme loop3(n)
  s = 0
  i = n
  while i > 0
    for j = 1 to i
      s = s + 1
    i = i − 1
```

```
Algoritme loop4(n)
  i = 0
  j = 0
  while i ≤ n
    if i < j then
      i = i + 1
    else
      j = j + 1
      i = 0
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

$\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n^2)$ $\Theta((\log n)^2)$ $\Theta(n \log n)$ $\Theta(n^2 \cdot \log n)$ $\Theta(n\sqrt{n})$ $\Theta(n)$

loop1

loop2

loop3

loop4

Opgave 13 (6 %) check

```
Algoritme loop1(n)
  s = 1
  for i = 1 to n
    s = s + 1
```

```
Algoritme loop2(n)
  i = 1
  while i ≤ n
    i = 2 * i
```

```
Algoritme loop3(n)
  i = 1
  while i * i ≤ n
    i = i + i
```

```
Algoritme loop4(n)
  i = 1
  while i ≤ n
    j = 0
    while j ≤ i
      j = j + 1
    i = 2 * i
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

$\Theta(n^3)$ $\Theta(\log n)$ $\Theta((\log n)^2)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(\frac{\log n}{\log \log n})$ $\Theta(n \log n)$ $\Theta(n^2)$

loop1

loop2

loop3

loop4

Opgave 14 (6 %) check

Algoritme loop1(n)
  for $i = 1$ **to** n
    $j = i$
    while $j > 0$
      $j = j - 1$

Algoritme loop2(n)
  $i = n$
  while $i > 0$
    $i = i - 1$

Algoritme loop3(n)
  $i = 1$
  while $i \leq n$
    $j = 1$
    while $j \leq i$
      $j = j + 1$
    $i = i + 1$

Algoritme loop4(n)
  $i = n$
  while $i > 0$
    $j = i$
    while $j > 0$
      $j = \lfloor j/2 \rfloor$
    $i = \lfloor i/2 \rfloor$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n^2 \cdot \log n)$	$\Theta(\sqrt[3]{n})$	$\Theta((\log n)^2)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$
loop1								
loop2								
loop3								
loop4								

Opgave 15 (6 %)

check

```
Algoritme loop1(n)        Algoritme loop2(n)
  s = 1                     s = 1
  while s ≤ n               for i = n to 1 step −1
    s = s + 1                 for j = n to 1 step −1
                                s = s + 1

Algoritme loop3(n)        Algoritme loop4(n)
  i = 1                     i = 1
  while i ≤ n ∗ n           j = 1
    i = 3 ∗ i               s = 0
                            while i ≤ n
                              if i = j then
                                for k = 1 to n
                                  s = s + 1
                                j = 2 ∗ j
                              i = i + 1
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n \log n)$	$\Theta((\log n)^2)$	$\Theta(\frac{\log n}{\log \log n})$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(n^2)$	$\Theta(\log n)$	$\Theta(n^2 \cdot \log n)$
loop1								
loop2								
loop3								
loop4								

Opgave 16 (6 %) check

```
Algoritme loop1(n)
  s = 0
  for i = 1 to n
    for j = 1 to n
      for k = 1 to n
        s = s + 1

Algoritme loop2(n)
  i = n
  while i > 0
    i = i - 1

Algoritme loop3(n)
  s = 0
  i = 1
  while s ≤ n
    s = s + i
    i = i + 1

Algoritme loop4(n)
  for i = 1 to n
    for j = 1 to i
      k = 1
      while k ≤ i + j
        k = 2 * k
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n^2 \cdot \log n)$	$\Theta(n^2)$	$\Theta(n \log n)$	$\Theta(\sqrt{n})$	$\Theta(n^3)$	$\Theta((\log n)^2)$	$\Theta(n)$	$\Theta(\log n)$
loop1								
loop2								
loop3								
loop4								

Opgave 17 (6 %) check

```
Algoritme loop1(n)
  s = 1
  for i = n to 1 step −1
    for j = n to 1 step −1
      s = s + 1

Algoritme loop2(n)
  for i = 1 to n
    j = i
    while j > 0
      j = j − 1

Algoritme loop3(n)
  i = 1
  while i * i ≤ n
    i = i + i

Algoritme loop4(n)
  for i = 1 to n
    j = 0
    while j ≤ n
      j = j + i
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta((\log n)^2)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n^2)$	$\Theta(\log \log n)$	$\Theta(n \log n)$	$\Theta(\log n)$	$\Theta(n)$
loop1								
loop2								
loop3								
loop4								

Opgave 18 (6 %) check

```
Algoritme loop1(n)        Algoritme loop2(n)
  s = 0                     s = 0
  for i = 1 to n            for i = 1 to n
      for j = 1 to i * i        for j = 1 to n
          s = s + 1                 for k = 1 to n
                                        s = s + 1

Algoritme loop3(n)        Algoritme loop4(n)
  i = 1                     i = n
  while i ≤ n               while i > 0
      j = 1                     j = i
      while j ≤ n               while j > 0
          j = 2 * j                 j = ⌊j/2⌋
      i = 2 * i                 i = ⌊i/2⌋
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

$\Theta(\log n)$ $\Theta((\log n)^2)$ $\Theta(\sqrt{n})$ $\Theta(n^2)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(\frac{\log n}{\log \log n})$ $\Theta(n^3)$

loop1

loop2

loop3

loop4

Opgave 19 (6 %) check

```
Algoritme loop1(n)        Algoritme loop2(n)
  for i = 1 to n            s = 1
      j = i                 for i = 1 to n
      while j > 0               for j = 1 to n
          j = j − 1                 s = s + 1

Algoritme loop3(n)        Algoritme loop4(n)
  i = 1                     i = n
  while i * i ≤ n           while i > 0
      i = i + i                 if i ulige then
                                    i = i − 1
                                else
                                    i = i/2
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

$\Theta(\sqrt{n})$ $\Theta(\log \log n)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(\log n)$ $\Theta((\log n)^2)$ $\Theta(n^3)$ $\Theta(n^2)$

loop1

loop2

loop3

loop4

Opgave 20 (6 %) check

Algoritme loop1(n)
  $s = 0$
  for $i = 1$ **to** n
    for $j = 1$ **to** $i * i$
      $s = s + 1$

Algoritme loop2(n)
  $s = 1$
  for $i = 1$ **to** n
    for $j = 1$ **to** n
      $s = s + 1$

Algoritme loop3(n)
  $i = 1$
  while $i * i \leq n$
    $i = i + i$

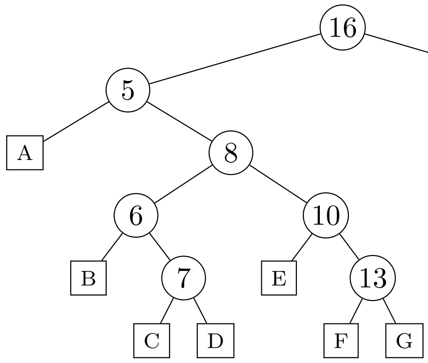
Algoritme loop4(n)
  $i = 1$
  while $i \leq n$
    $j = 0$
    while $j \leq i$
      $j = j + 1$
    $i = 2 * i$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta((\log n)^2)$	$\Theta(n^3)$	$\Theta(\log n)$	$\Theta(n \log n)$	$\Theta(\frac{\log n}{\log \log n})$	$\Theta(n)$	$\Theta(n^2)$	$\Theta(\sqrt{n})$
loop1								
loop2								
loop3								
loop4								

Indsættelser i søgetræer

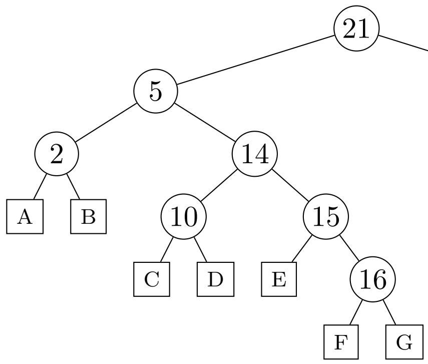
Opgave 21 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 12, 25, 11, 14 og 29 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(12)											
INSERT(25)											
INSERT(11)											
INSERT(14)											
INSERT(29)											

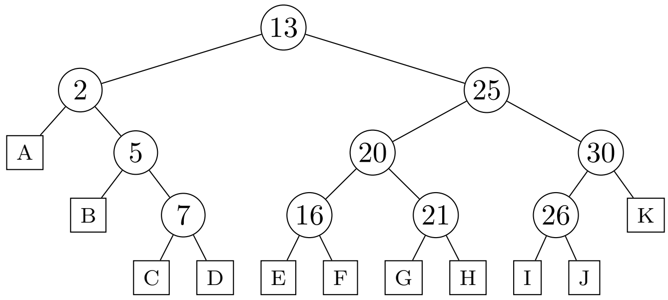
Opgave 22 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 1, 12, 9, 3 og 7 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(1)											
INSERT(12)											
INSERT(9)											
INSERT(3)											
INSERT(7)											

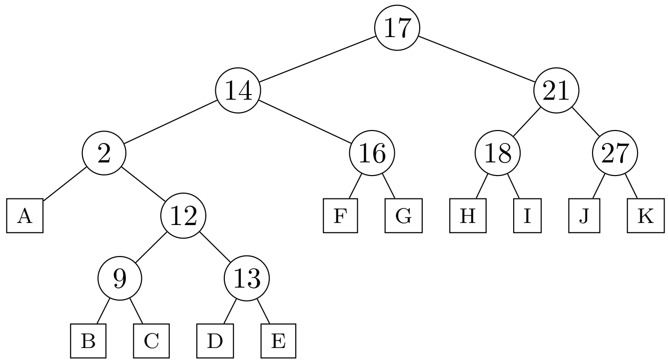
Opgave 23 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 17, 18, 29, 23 og 27 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
Insert(17)											
Insert(18)											
Insert(29)											
Insert(23)											
Insert(27)											

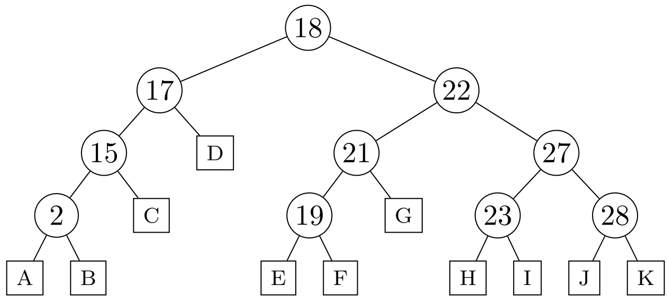
Opgave 24 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 22, 3, 26, 24 og 1 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
Insert(22)											
Insert(3)											
Insert(26)											
Insert(24)											
Insert(1)											

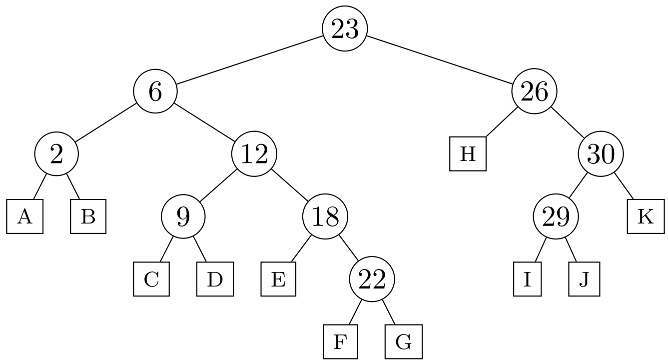
Opgave 25 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 7, 29, 8, 4 og 6 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(7)											
INSERT(29)											
INSERT(8)											
INSERT(4)											
INSERT(6)											

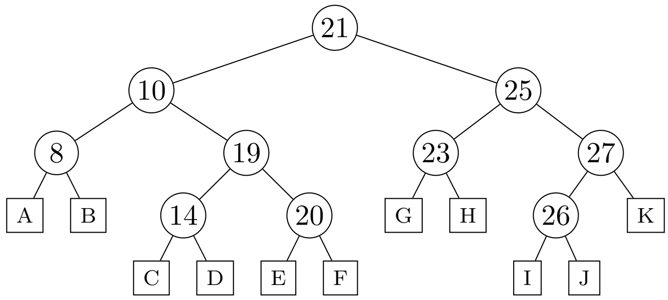
Opgave 26 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 31, 16, 4, 20 og 19 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(31)											
INSERT(16)											
INSERT(4)											
INSERT(20)											
INSERT(19)											

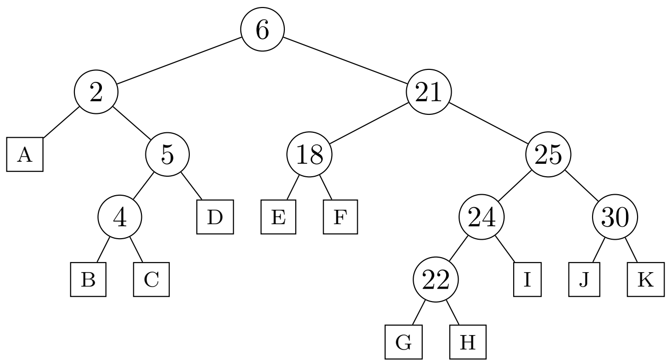
Opgave 27 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 15, 11, 13, 16 og 7 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(15)											
INSERT(11)											
INSERT(13)											
INSERT(16)											
INSERT(7)											

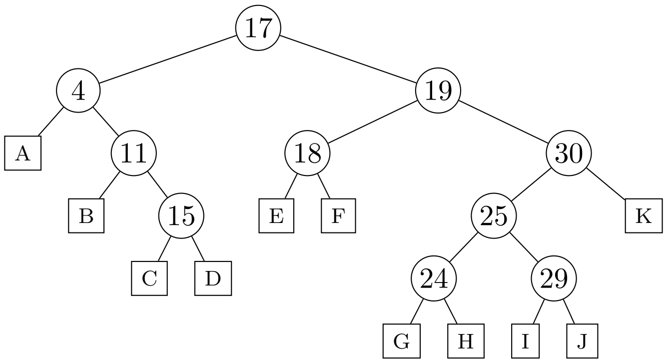
Opgave 28 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 15, 3, 10, 16 og 12 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(15)											
INSERT(3)											
INSERT(10)											
INSERT(16)											
INSERT(12)											

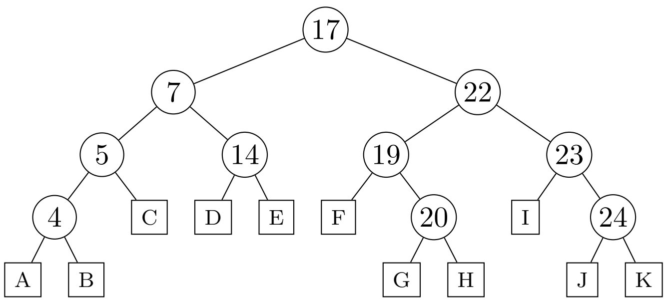
Opgave 29 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 14, 26, 23, 16 og 6 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
Insert(14)											
Insert(26)											
Insert(23)											
Insert(16)											
Insert(6)											

Opgave 30 (4 %) check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 13, 16, 8, 10 og 15 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
Insert(13)											
Insert(16)											
Insert(8)											
Insert(10)											
Insert(15)											

Max-Heap-Insert

Opgave 31 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 4, 7, 11, 3, 13, 9 og 5 i den givne rækkefølge med Max-Heap-Insert, startende med den tomme heap.

1	2	3	4	5	6	7
13	7	11	3	4	9	5

1	2	3	4	5	6	7
13	11	9	3	4	7	5

1	2	3	4	5	6	7
4	7	11	3	13	9	5

1	2	3	4	5	6	7
13	11	9	7	5	4	3

1	2	3	4	5	6	7
11	13	9	3	7	4	5

Opgave 32 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 6, 7, 12, 11, 9, 10 og 13 i den givne rækkefølge med Max-Heap-Insert, startende med den tomme heap.

1	2	3	4	5	6	7
6	7	12	11	9	10	13

1	2	3	4	5	6	7
13	11	12	7	9	10	6

1	2	3	4	5	6	7
13	12	11	10	9	7	6

1	2	3	4	5	6	7
12	11	13	7	9	10	6

1	2	3	4	5	6	7
13	11	12	6	9	7	10

Opgave 33 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 13, 4, 5, 6, 14, 3 og 10 i den givne rækkefølge med Max-Heap-Insert, startende med den tomme heap.

1	2	3	4	5	6	7
14	13	10	6	5	4	3

1	2	3	4	5	6	7
14	13	10	4	6	3	5

1	2	3	4	5	6	7
14	13	10	6	4	3	5

1	2	3	4	5	6	7
13	4	5	6	14	3	10

1	2	3	4	5	6	7
13	14	10	6	4	3	5

Opgave 34 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 5, 7, 8, 14, 2, 4 og 9 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
5	7	8	14	2	4	9

1	2	3	4	5	6	7
14	7	9	5	2	4	8

1	2	3	4	5	6	7
14	9	8	7	5	4	2

1	2	3	4	5	6	7
14	8	9	5	2	4	7

1	2	3	4	5	6	7
8	14	9	7	2	4	5

Opgave 35 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 4, 2, 11, 14, 6, 5 og 8 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
14	11	8	6	5	4	2

1	2	3	4	5	6	7
4	2	11	14	6	5	8

1	2	3	4	5	6	7
14	6	11	2	4	5	8

1	2	3	4	5	6	7
14	11	8	2	6	4	5

1	2	3	4	5	6	7
11	14	8	2	6	5	4

Opgave 36 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 4, 3, 7, 8, 5, 10 og 11 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
11	7	10	3	5	4	8

1	2	3	4	5	6	7
11	10	8	7	5	4	3

1	2	3	4	5	6	7
4	3	7	8	5	10	11

1	2	3	4	5	6	7
11	8	10	3	5	4	7

1	2	3	4	5	6	7
7	8	11	3	5	10	4

Opgave 37 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 8, 9, 10, 13, 6, 5 og 12 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
13	10	12	8	6	5	9

1	2	3	4	5	6	7
13	9	12	8	6	5	10

1	2	3	4	5	6	7
8	9	10	13	6	5	12

1	2	3	4	5	6	7
13	12	10	9	8	6	5

1	2	3	4	5	6	7
10	13	12	9	6	5	8

Opgave 38 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 2, 6, 1, 4, 7, 9 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
2	6	1	4	7	9	14

1	2	3	4	5	6	7
14	6	9	2	4	1	7

1	2	3	4	5	6	7
14	9	7	6	4	2	1

1	2	3	4	5	6	7
6	7	14	4	2	9	1

1	2	3	4	5	6	7
14	7	9	4	6	2	1

Opgave 39 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 1, 10, 9, 8, 11, 14 og 3 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
1	10	9	8	11	14	3

1	2	3	4	5	6	7
14	10	11	1	8	9	3

1	2	3	4	5	6	7
14	11	9	8	10	1	3

1	2	3	4	5	6	7
10	11	14	8	1	9	3

1	2	3	4	5	6	7
14	11	10	9	8	3	1

Opgave 40 (4 %) check

Angiv den binære max-heap efter indsættelse af elementerne 11, 12, 9, 2, 4, 13 og 10 i den givne rækkefølge med Max-Heap-Insert, startende med den tomme heap.

1	2	3	4	5	6	7
13	12	11	2	4	9	10

1	2	3	4	5	6	7
12	11	13	2	4	9	10

1	2	3	4	5	6	7
13	11	12	2	4	9	10

1	2	3	4	5	6	7
13	12	11	10	9	4	2

1	2	3	4	5	6	7
11	12	9	2	4	13	10

Build-Max-Heap

Opgave 41 (4 %) check

1	2	3	4	5	6	7	8	9
7	6	5	8	9	3	4	1	2

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
7	9	5	8	6	3	4	1	2

1	2	3	4	5	6	7	8	9
9	8	5	7	6	3	4	1	2

1	2	3	4	5	6	7	8	9
9	8	5	6	7	3	4	1	2

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

Opgave 42 (4 %)

check

1	2	3	4	5	6	7	8	9
3	1	5	2	8	4	6	7	9

Hvad er resultat af BUILD-MAX-HEAP på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	6	7	3	4	5	1	2

1	2	3	4	5	6	7	8	9
9	8	6	7	2	3	4	1	5

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
5	8	6	9	1	4	3	7	2

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

Opgave 43 (4 %)

check

1	2	3	4	5	6	7	8	9
1	8	5	9	3	4	2	6	7

Hvad er resultat af BUILD-MAX-HEAP på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
8	9	5	7	3	4	2	6	1

1	2	3	4	5	6	7	8	9
9	8	5	7	3	4	2	6	1

1	2	3	4	5	6	7	8	9
9	8	5	7	3	4	2	1	6

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

Opgave 44 (4 %) check

1	2	3	4	5	6	7	8	9
6	7	1	8	3	4	9	2	5

Hvad er resultat af BUILD-MAX-HEAP på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	7	8	6	3	1	4	2	5

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
7	8	9	6	3	4	1	2	5

1	2	3	4	5	6	7	8	9
9	8	6	7	3	4	1	2	5

Opgave 45 (4 %) check

1	2	3	4	5	6	7	8	9
5	6	7	4	2	9	8	1	3

Hvad er resultat af BUILD-MAX-HEAP på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
7	6	9	4	2	5	8	1	3

1	2	3	4	5	6	7	8	9
9	5	8	4	2	6	7	1	3

1	2	3	4	5	6	7	8	9
9	6	8	4	2	7	5	1	3

Opgave 46 (4 %) check

1	2	3	4	5	6	7	8	9
7	3	9	4	2	6	1	8	5

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
9	8	7	5	2	6	1	4	3

1	2	3	4	5	6	7	8	9
9	4	7	8	2	6	1	3	5

1	2	3	4	5	6	7	8	9
9	8	7	5	2	6	1	3	4

Opgave 47 (4 %) check

1	2	3	4	5	6	7	8	9
8	1	2	7	3	6	9	5	4

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
8	7	9	5	3	6	2	1	4

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
9	7	8	5	3	2	6	1	4

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
9	7	8	5	3	6	2	1	4

Opgave 48 (4 %) check

1	2	3	4	5	6	7	8	9
1	2	7	6	3	5	4	8	9

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
9	8	5	7	3	2	4	1	6

1	2	3	4	5	6	7	8	9
7	6	5	9	3	1	4	8	2

1	2	3	4	5	6	7	8	9
9	8	7	6	3	5	4	2	1

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

Opgave 49 (4 %) check

1	2	3	4	5	6	7	8	9
3	4	6	8	9	1	7	2	5

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

1	2	3	4	5	6	7	8	9
9	8	7	5	4	1	6	2	3

1	2	3	4	5	6	7	8	9
6	9	7	8	4	1	3	2	5

1	2	3	4	5	6	7	8	9
9	8	7	5	6	1	4	2	3

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

Opgave 50 (4 %) check

1	2	3	4	5	6	7	8	9
2	5	6	7	9	3	4	1	8

Hvad er resultat af Build-Max-Heap på ovenstående array ?

1	2	3	4	5	6	7	8	9
9	8	6	7	5	3	4	1	2

1	2	3	4	5	6	7	8	9
9	8	5	7	6	3	4	1	2

1	2	3	4	5	6	7	8	9
6	9	4	8	5	3	2	1	7

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

Heap-Extract-Max

Opgave 51 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	20	22	15	16	4	7	11	2	3	13	12

Hvad er resultat af Heap-Extract-Max på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
24	22	20	11	15	16	4	7	2	3	13	12

1	2	3	4	5	6	7	8	9	10	11	12
24	22	20	12	15	16	4	7	11	2	3	13

1	2	3	4	5	6	7	8	9	10	11	12
24	22	20	11	15	16	4	7	12	2	3	13

1	2	3	4	5	6	7	8	9	10	11	12	13
24	22	20	11	15	16	4	7		2	3	13	12

1	2	3	4	5	6	7	8	9	10	11	12
24	20	22	15	16	12	7	11	2	3	13	4

Opgave 52 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
24	23	21	18	22	19	7	4	13	6	11	17	16

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
23	22	21	18	16	19	7	4	13	6	11	17

1	2	3	4	5	6	7	8	9	10	11	12	13
23	22	21	18	11	19	7	4	13	6		17	16

1	2	3	4	5	6	7	8	9	10	11	12
23	22	21	18	11	19	7	4	13	6	17	16

1	2	3	4	5	6	7	8	9	10	11	12
23	22	21	18	11	19	7	4	13	6	16	17

1	2	3	4	5	6	7	8	9	10	11	12
23	22	18	21	19	16	4	13	6	11	17	7

Opgave 53 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
23	21	19	18	14	9	13	3	5	6	12	8	7

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
21	18	19	7	14	9	13	3	5	6	12	8

1	2	3	4	5	6	7	8	9	10	11	12
21	19	18	14	12	13	3	5	6	9	8	7

1	2	3	4	5	6	7	8	9	10	11	12	13
21	18	19	5	14	9	13	3		6	12	8	7

1	2	3	4	5	6	7	8	9	10	11	12
21	18	19	5	14	9	13	3	7	6	12	8

1	2	3	4	5	6	7	8	9	10	11	12
21	18	19	5	14	9	13	3	6	12	8	7

Opgave 54 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	23	16	22	21	17	10	8	13	14	6	18

Hvad er resultat af Heap-Extract-Max på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12	13
25	22	23	16	14	21	17	10	8	13		6	18

1	2	3	4	5	6	7	8	9	10	11	12
25	22	23	16	14	21	17	10	8	13	6	18

1	2	3	4	5	6	7	8	9	10	11	12
25	22	23	16	14	21	17	10	8	13	18	6

1	2	3	4	5	6	7	8	9	10	11	12
25	23	18	22	21	17	10	8	13	14	6	16

1	2	3	4	5	6	7	8	9	10	11	12
25	22	23	16	18	21	17	10	8	13	14	6

Opgave 55 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	20	21	17	18	13	2	1	7	8	16	12

Hvad er resultat af Heap-Extract-Max på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
24	20	21	17	18	13	2	1	7	8	16	12

1	2	3	4	5	6	7	8	9	10	11	12
24	21	20	12	17	18	13	2	1	7	8	16

1	2	3	4	5	6	7	8	9	10	11	12	13
24	21	20	2	17	18	13		1	7	8	16	12

1	2	3	4	5	6	7	8	9	10	11	12
24	21	20	2	17	18	13	1	7	8	16	12

1	2	3	4	5	6	7	8	9	10	11	12
24	21	20	2	17	18	13	12	1	7	8	16

Opgave 56 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	23	22	6	7	18	13	3	2	4	5	9	12

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
23	22	13	7	18	12	3	2	4	5	9	6

1	2	3	4	5	6	7	8	9	10	11	12
23	7	22	6	5	18	13	3	2	4	12	9

1	2	3	4	5	6	7	8	9	10	11	12
23	7	22	6	5	18	13	3	2	4	9	12

1	2	3	4	5	6	7	8	9	10	11	12
23	12	22	6	7	18	13	3	2	4	5	9

1	2	3	4	5	6	7	8	9	10	11	12	13
23	7	22	6	5	18	13	3	2	4		9	12

Opgave 57 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	24	19	18	16	6	4	8	10	13	1	15

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
25	19	24	8	18	16	6	4	10	13	1	15

1	2	3	4	5	6	7	8	9	10	11	12
25	24	19	18	16	15	4	8	10	13	1	6

1	2	3	4	5	6	7	8	9	10	11	12	13
25	19	24	8	18	16	6	4		10	13	1	15

1	2	3	4	5	6	7	8	9	10	11	12
25	19	24	15	18	16	6	4	8	10	13	1

1	2	3	4	5	6	7	8	9	10	11	12
25	19	24	8	18	16	6	4	15	10	13	1

Opgave 58 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	23	17	20	19	14	12	7	1	9	2	4	10

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
23	20	17	7	19	14	12	1	9	2	4	10

1	2	3	4	5	6	7	8	9	10	11	12
23	20	17	10	19	14	12	7	1	9	2	4

1	2	3	4	5	6	7	8	9	10	11	12
23	19	20	17	14	12	7	1	9	2	4	10

1	2	3	4	5	6	7	8	9	10	11	12	13
23	20	17	7	19	14	12		1	9	2	4	10

1	2	3	4	5	6	7	8	9	10	11	12
23	20	17	7	19	14	12	10	1	9	2	4

Opgave 59 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	18	17	19	14	3	15	2	10	5	4	13

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
24	19	18	17	13	14	3	15	2	10	5	4

1	2	3	4	5	6	7	8	9	10	11	12	13
24	19	18	17	10	14	3	15	2		5	4	13

1	2	3	4	5	6	7	8	9	10	11	12
24	19	18	17	10	14	3	15	2	5	4	13

1	2	3	4	5	6	7	8	9	10	11	12
24	19	17	18	14	13	15	2	10	5	4	3

1	2	3	4	5	6	7	8	9	10	11	12
24	19	18	17	10	14	3	15	2	13	5	4

Opgave 60 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	23	15	22	16	13	10	1	5	3	9	7	6

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
23	22	15	5	16	13	10	1	3	9	7	6

1	2	3	4	5	6	7	8	9	10	11	12
23	16	22	15	13	10	1	5	3	9	7	6

1	2	3	4	5	6	7	8	9	10	11	12
23	22	15	6	16	13	10	1	5	3	9	7

1	2	3	4	5	6	7	8	9	10	11	12
23	22	15	5	16	13	10	1	6	3	9	7

1	2	3	4	5	6	7	8	9	10	11	12	13
23	22	15	5	16	13	10	1		3	9	7	6

Partition

Opgave 61 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	6	12	16	15	27	13	5	21	25	1	19	2	17	10

Angiv resultatet af at anvende PARTITION$(A, 2, 12)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	6	12	16	15	13	5	1	19	27	21	25	2	17	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	5	6	10	12	13	15	16	17	19	21	25	27	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	6	12	16	15	13	5	1	19	25	27	21	2	17	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	1	5	6	12	13	15	16	19	21	25	27	2	17	10

Opgave 62 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	19	6	30	15	25	10	8	28	1	24	29	22	17	12

Angiv resultatet af at anvende PARTITION$(A, 4, 14)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	19	6	15	10	8	1	17	28	30	24	29	22	25	12

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	19	6	15	10	8	1	17	30	25	28	24	29	22	12

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	6	8	10	11	12	15	17	19	22	24	25	28	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	19	6	1	8	10	15	17	22	24	25	28	29	30	12

Opgave 63 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	17	2	25	27	30	7	10	26	11	4	24	9	29	16

Angiv resultatet af at anvende PARTITION$(A, 3, 13)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	17	2	7	4	9	25	27	30	10	26	11	24	29	16

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	7	9	10	11	16	17	23	24	25	26	27	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	17	2	7	4	9	25	10	26	11	27	24	30	29	16

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	17	2	4	7	9	10	11	24	25	26	27	30	29	16

Opgave 64 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	14	12	11	18	22	25	10	7	13	24	1	5	3	9

Angiv resultatet af at anvende PARTITION$(A, 3, 14)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	14	1	3	5	7	10	11	12	13	18	22	24	25	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	5	7	9	10	11	12	13	14	16	18	22	24	25

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	14	1	3	12	11	18	22	25	10	7	13	24	5	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	14	1	3	18	22	25	10	7	13	24	12	5	11	9

Opgave 65 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	27	28	22	6	13	30	20	19	23	24	21	25	29	12

Angiv resultatet af at anvende PARTITION$(A, 2, 13)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	6	13	19	20	21	22	23	24	25	27	28	30	29	12

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	22	6	13	20	19	23	24	21	25	28	27	30	29	12

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	12	13	17	19	20	21	22	23	24	25	27	28	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	22	6	13	20	19	23	24	21	25	27	28	30	29	12

Opgave 66 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	11	9	5	27	24	20	4	2	19	23	22	8	29	21

Angiv resultatet af at anvende PARTITION$(A, 2, 13)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	5	4	2	8	24	20	9	11	19	23	22	27	29	21

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	8	9	11	19	20	21	22	23	24	27	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	2	4	5	8	9	11	19	20	22	23	24	27	29	21

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	5	4	2	8	11	9	27	24	20	19	23	22	29	21

Opgave 67 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	20	11	17	3	12	1	22	19	8	21	23	9	13	18

Angiv resultatet af at anvende PARTITION$(A, 3, 14)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	20	11	3	12	1	8	9	13	17	21	23	22	19	18

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	20	1	3	8	9	11	12	13	17	19	21	22	23	18

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	6	8	9	11	12	13	17	18	19	20	21	22	23

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	20	11	3	12	1	8	9	13	17	22	19	21	23	18

Opgave 68 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	22	5	30	12	11	26	4	10	18	14	16	13	1	9

Angiv resultatet af at anvende Partition($A, 2, 12$) på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	5	7	9	10	11	12	13	14	16	18	22	26	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	4	5	10	11	12	14	16	18	22	26	30	13	1	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	5	12	11	4	10	14	16	30	18	26	22	13	1	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	5	12	11	4	10	14	16	22	30	26	18	13	1	9

Opgave 69 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	16	22	30	12	20	17	15	29	4	8	5	27	24	23

Angiv resultatet af at anvende Partition($A, 2, 14$) på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	5	8	9	12	15	16	17	20	22	23	24	27	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	16	22	12	20	17	15	4	8	5	24	30	27	29	23

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	16	22	12	20	17	15	4	8	5	24	30	29	27	23

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	4	5	8	12	15	16	17	20	22	24	27	29	30	23

Opgave 70 (4 %) check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	17	18	7	27	5	14	8	29	16	10	25	3	15	28

Angiv resultatet af at anvende Partition($A, 4, 12$) på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	17	18	7	5	14	8	16	10	25	27	29	3	15	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	17	18	5	7	8	10	14	16	25	27	29	3	15	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	5	7	8	10	14	15	16	17	18	25	27	28	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	17	18	7	5	14	8	16	10	25	29	27	3	15	28

Radix-sort

Opgave 71 (4 %) check

3441 0022 2322 2041 2022 2120

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

2120	0022	2322	2022	3441	2041
2120	3441	2041	0022	2322	2022
2120	0022	2022	2322	2041	3441
0022	2041	2022	2120	2322	3441
0022	2022	2041	2120	2322	3441

Opgave 72 (4 %) check

0412 0040 1040 1212 0240 0011

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

0011	0412	1212	0040	0240	1040
0011	0040	0240	0412	1040	1212
0011	0412	1212	0040	1040	0240
0040	1040	0240	0011	0412	1212
0040	0011	0240	0412	1040	1212

Opgave 73 (4 %) check

4034 1434 3220 3411 1420 4311

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

1420	1434	3220	3411	4034	4311
3411	4311	3220	1420	4034	1434
3220	1420	3411	4311	4034	1434
1434	1420	3220	3411	4034	4311
3411	4311	1420	3220	1434	4034

Opgave 74 (4 %) check

0320 4301 2001 0301 0010 1101

Betragt Radix-Sort anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at Radix-Sort har sorteret tallene efter de *to* mindst betydende cifre.

0010	0301	0320	1101	2001	4301
0301	1101	2001	4301	0010	0320
4301	2001	0301	1101	0010	0320
0010	0320	4301	2001	0301	1101
0010	0320	0301	1101	2001	4301

Opgave 75 (4 %) check

2224 3224 0042 3221 3324 0021

Betragt Radix-Sort anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at Radix-Sort har sorteret tallene efter de *to* mindst betydende cifre.

0042	0021	2224	3224	3221	3324
3221	0021	0042	2224	3224	3324
0021	0042	2224	3221	3224	3324
0021	3221	2224	3224	3324	0042
3221	0021	2224	3224	3324	0042

Opgave 76 (4 %) check

1412 3333 3324 2424 4133 4324

Betragt Radix-Sort anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at Radix-Sort har sorteret tallene efter de *to* mindst betydende cifre.

1412	3324	2424	4324	3333	4133
1412	2424	3324	4324	3333	4133
1412	2424	3333	3324	4133	4324
1412	3333	4133	3324	2424	4324
1412	2424	3324	3333	4133	4324

Opgave 77 (4 %) check

1123 1232 3332 4141 2341 4123

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

1123 4123 1232 3332 4141 2341

4141 2341 1232 3332 1123 4123

1123 4123 1232 3332 2341 4141

1123 1232 2341 3332 4123 4141

1123 1232 2341 3332 4141 4123

Opgave 78 (4 %) check

2214 4121 3021 2012 3014 2121

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

2012 2214 3014 2121 3021 4121

2012 2121 2214 3014 3021 4121

2012 2121 2214 3021 3014 4121

4121 3021 2121 2012 2214 3014

2012 2214 3014 4121 3021 2121

Opgave 79 (4 %) check

2041 1213 0020 0113 2020 1041

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

0020 0113 1041 1213 2041 2020

1213 0113 0020 2020 2041 1041

0020 0113 1041 1213 2020 2041

0113 1213 0020 2020 1041 2041

0020 2020 2041 1041 1213 0113

Opgave 80 (4 %) check

2331 0242 2442 4312 2231 0212

Betragt Radix-Sort anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at Radix-Sort har sorteret tallene efter de *to* mindst betydende cifre.

0212	4312	2231	2331	0242	2442
0242	0212	2231	2331	2442	4312
4312	0212	2331	2231	0242	2442
2331	2231	4312	0212	0242	2442
0212	0242	2231	2331	2442	4312

Lineær probing

Opgave 81 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
16			7				9		21	5

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 2k \bmod 11$.

Angiv positionerne de fem elementer 3, 4, 6, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 7, 9, 16 og 21).

	0	1	2	3	4	5	6	7	8	9	10
Insert(3)											
Insert(4)											
Insert(6)											
Insert(10)											
Insert(11)											

Opgave 82 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
22		19	8		20					18

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \bmod 11$.

Angiv positionerne de fem elementer 0, 5, 6, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 18, 19, 20 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(5)											
INSERT(6)											
INSERT(7)											
INSERT(9)											

Opgave 83 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
0	22							13	5	19

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 4k \bmod 11$.

Angiv positionerne de fem elementer 2, 3, 4, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 5, 13, 19 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(2)											
INSERT(3)											
INSERT(4)											
INSERT(9)											
INSERT(10)											

Opgave 84 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
7			12					10	21	18

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \bmod 11$.

Angiv positionerne de fem elementer 0, 2, 3, 4 og 6 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 7, 10, 12, 18 og 21).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(2)											
INSERT(3)											
INSERT(4)											
INSERT(6)											

Opgave 85 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
0	9	20							15	13

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 5k \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 6, 8 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 9, 13, 15 og 20).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(6)											
INSERT(8)											
INSERT(11)											

Opgave 86 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
						2	13	21	3	7

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \bmod 11$.

Angiv positionerne de fem elementer 0, 1, 5, 6 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 3, 7, 13 og 21).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(1)											
INSERT(5)											
INSERT(6)											
INSERT(10)											

Opgave 87 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
			16		12			6	17	13

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 5k \bmod 11$.

Angiv positionerne de fem elementer 2, 3, 5, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 12, 13, 16 og 17).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(2)											
INSERT(3)											
INSERT(5)											
INSERT(9)											
INSERT(11)											

Opgave 88 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
22	0				12	21				13

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 5k \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 3, 6 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 12, 13, 21 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(3)											
INSERT(6)											
INSERT(8)											

Opgave 89 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
		18	5	3					4	15

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 5k \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 7, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 4, 5, 15 og 18).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(7)											
INSERT(8)											
INSERT(9)											

Opgave 90 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
11						13		21	14	3

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \bmod 11$.

Angiv positionerne de fem elementer 0, 4, 6, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 11, 13, 14 og 21).

	0	1	2	3	4	5	6	7	8	9	10
Insert(0)											
Insert(4)											
Insert(6)											
Insert(8)											
Insert(10)											

Kvadratisk probing

Opgave 91 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
0		18	16						15	22

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 5k \bmod 11$ og $h(k, i) = (h'(k) + 5i + 5i^2) \bmod 11$.

Angiv positionerne de fem elementer 4, 5, 6, 7 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 15, 16, 18 og 22).

	0	1	2	3	4	5	6	7	8	9	10
Insert(4)											
Insert(5)											
Insert(6)											
Insert(7)											
Insert(11)											

Opgave 92 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
			6		12	21		17	15	

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 5k \bmod 11$ og $h(k, i) = (h'(k) + 2i + 4i^2) \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 4, 5 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 12, 15, 17 og 21).

	0	1	2	3	4	5	6	7	8	9	10
Insert(1)											
Insert(2)											
Insert(4)											
Insert(5)											
Insert(11)											

Opgave 93 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
22							11	21	3	7

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 3k \bmod 11$ og $h(k, i) = (h'(k) + 2i + 5i^2) \bmod 11$.

Angiv positionerne de fem elementer 0, 2, 5, 6 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 7, 11, 21 og 22).

	0	1	2	3	4	5	6	7	8	9	10
Insert(0)											
Insert(2)											
Insert(5)											
Insert(6)											
Insert(10)											

Opgave 94 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
11	20				19		8			9

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 5k \bmod 11$ og $h(k,i) = (h'(k) + 4i + 5i^2) \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 3, 4 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 9, 11, 19 og 20).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(3)											
INSERT(4)											
INSERT(7)											

Opgave 95 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
22				2	4			15	21	

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 2k \bmod 11$ og $h(k,i) = (h'(k) + 3i + 5i^2) \bmod 11$.

Angiv positionerne de fem elementer 0, 3, 5, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 4, 15, 21 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(3)											
INSERT(5)											
INSERT(9)											
INSERT(10)											

Opgave 96 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
			21		8			4	10	19

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 2k \bmod 11$ og $h(k,i) = (h'(k) + 2i + 3i^2) \bmod 11$.

Angiv positionerne de fem elementer 3, 5, 6, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 8, 10, 19 og 21).

	0	1	2	3	4	5	6	7	8	9	10
Insert(3)											
Insert(5)											
Insert(6)											
Insert(7)											
Insert(9)											

Opgave 97 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
			16	14	12		1			13

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 5k \bmod 11$ og $h(k,i) = (h'(k) + 2i^2) \bmod 11$.

Angiv positionerne de fem elementer 2, 5, 6, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 12, 13, 14 og 16).

	0	1	2	3	4	5	6	7	8	9	10
Insert(2)											
Insert(5)											
Insert(6)											
Insert(9)											
Insert(10)											

Opgave 98 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
				5	20		9		17	7

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 3k \bmod 11$ og $h(k,i) = (h'(k) + 2i^2) \bmod 11$.

Angiv positionerne de fem elementer 2, 3, 6, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 7, 9, 17 og 20).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(2)											
INSERT(3)											
INSERT(6)											
INSERT(8)											
INSERT(10)											

Opgave 99 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
	14			13			10	2	5	

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 4k \bmod 11$ og $h(k,i) = (h'(k) + 5i + 2i^2) \bmod 11$.

Angiv positionerne de fem elementer 0, 1, 3, 7 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 5, 10, 13 og 14).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(1)											
INSERT(3)											
INSERT(7)											
INSERT(8)											

Opgave 100 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
22			20			11	21		9	

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 4k \bmod 11$ og $h(k,i) = (h'(k) + 4i + 2i^2) \bmod 11$.

Angiv positionerne de fem elementer 1, 4, 6, 7 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 9, 11, 20, 21 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(4)											
INSERT(6)											
INSERT(7)											
INSERT(10)											

Dobbelt hashing

Opgave 101 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
	20		16	14					4	5

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 5k \bmod 11$ og $h_2(k) = 1 + (5k \bmod 10)$.

Angiv positionerne de fem elementer 0, 2, 3, 6 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 5, 14, 16 og 20).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(2)											
INSERT(3)											
INSERT(6)											
INSERT(8)											

Opgave 102 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
16				2		3	9			13

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 2k \bmod 11$ og $h_2(k) = 1 + (5k \bmod 10)$.

Angiv positionerne de fem elementer 1, 5, 6, 8 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 3, 9, 13 og 16).

	0	1	2	3	4	5	6	7	8	9	10
Insert(1)											
Insert(5)											
Insert(6)											
Insert(8)											
Insert(11)											

Opgave 103 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
11			12		9				22	18

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 3k \bmod 11$ og $h_2(k) = 1 + (4k \bmod 10)$.

Angiv positionerne de fem elementer 1, 3, 4, 5 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 9, 11, 12, 18 og 22).

	0	1	2	3	4	5	6	7	8	9	10
Insert(1)											
Insert(3)											
Insert(4)											
Insert(5)											
Insert(7)											

Opgave 104 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
0					9		6	21	10	

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 3k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 1, 2, 3, 4 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 6, 9, 10 og 21).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(3)											
INSERT(4)											
INSERT(11)											

Opgave 105 (4 %) check

0	1	2	3	4	5	6	7	8	9	10
	9	20	5	14					15	

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 5k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 1, 2, 3, 6 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 9, 14, 15 og 20).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(2)											
INSERT(3)											
INSERT(6)											
INSERT(7)											

Opgave 106 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
22				12	1				15	13

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 5k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 0, 3, 4, 5 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 12, 13, 15 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(3)											
INSERT(4)											
INSERT(5)											
INSERT(7)											

Opgave 107 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
		6	17			18	10	13		

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 4k \bmod 11$ og $h_2(k) = 1 + (5k \bmod 10)$.

Angiv positionerne de fem elementer 1, 3, 4, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 10, 13, 17 og 18).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(3)											
INSERT(4)											
INSERT(7)											
INSERT(9)											

Opgave 108 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
22		8	11						19	18

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 3k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 1, 3, 6, 7 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 11, 18, 19 og 22).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(1)											
INSERT(3)											
INSERT(6)											
INSERT(7)											
INSERT(10)											

Opgave 109 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11		21		2					10	16

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 2k \bmod 11$ og $h_2(k) = 1 + (3k \bmod 10)$.

Angiv positionerne de fem elementer 0, 4, 5, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 10, 11, 16 og 21).

	0	1	2	3	4	5	6	7	8	9	10
INSERT(0)											
INSERT(4)											
INSERT(5)											
INSERT(8)											
INSERT(9)											

Opgave 110 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
		1	18		19		7	15		

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 2k \bmod 11$ og $h_2(k) = 1 + (3k \bmod 10)$.

Angiv positionerne de fem elementer 2, 6, 8, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 7, 15, 18 og 19).

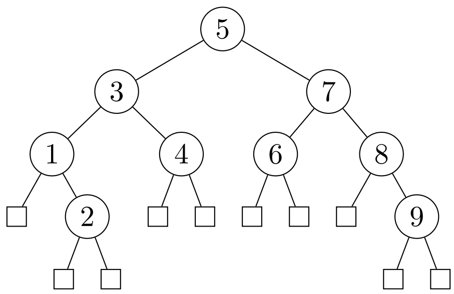
	0	1	2	3	4	5	6	7	8	9	10
Insert(2)											
Insert(6)											
Insert(8)											
Insert(9)											
Insert(11)											

Rød-sort træ

Opgave 111 (4 %)

check

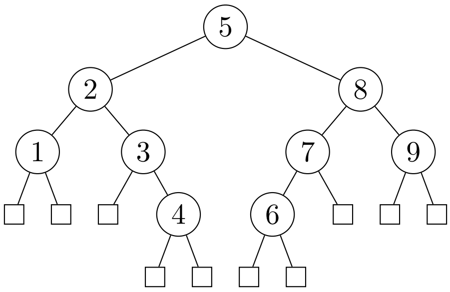
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
2, 5, 9		
2, 3, 7, 9		
2, 3, 6, 8, 9		
2, 9		
1, 2, 4, 7, 9		

Opgave 112 (4 %) check

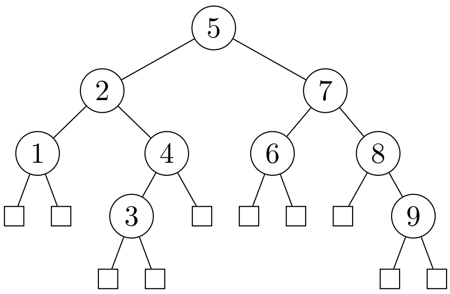
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
2, 4, 6, 7, 9		
1, 3, 4, 6, 8		
2, 4, 6, 8		
4, 6		
4, 5, 6		

Opgave 113 (4 %) check

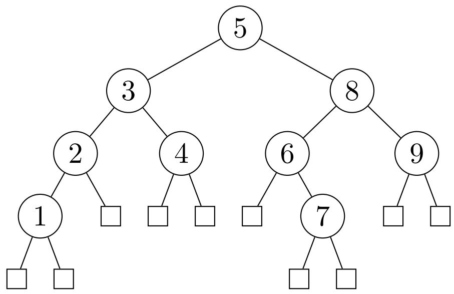
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
3, 9		
1, 3, 4, 7, 9		
2, 3, 6, 8, 9		
2, 3, 7, 9		
3, 5, 9		

Opgave 114 (4 %) check

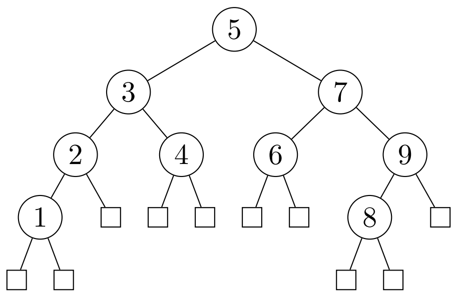
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
1, 2, 4, 7, 8		
1, 3, 7, 8		
1, 3, 6, 7, 9		
1, 7		
1, 5, 7		

Opgave 115 (4 %) check

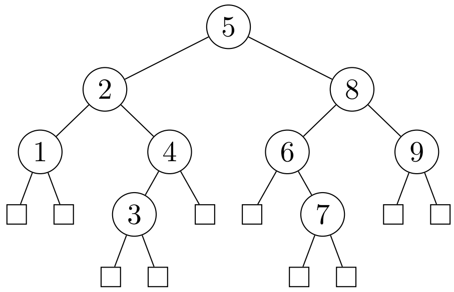
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
1, 2, 4, 7, 8		
1, 3, 6, 8, 9		
1, 5, 8		
1, 3, 7, 8		
1, 8		

Opgave 116 (4 %) check

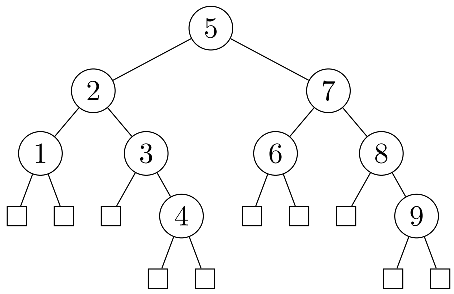
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
3, 7		
2, 3, 6, 7, 9		
1, 3, 4, 7, 8		
3, 5, 7		
2, 3, 7, 8		

Opgave 117 (4 %) check

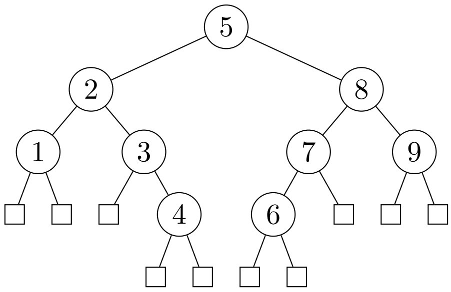
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
4, 9		
4, 5, 9		
1, 3, 4, 7, 9		
2, 4, 7, 9		
2, 4, 6, 8, 9		

Opgave 118 (4 %) check

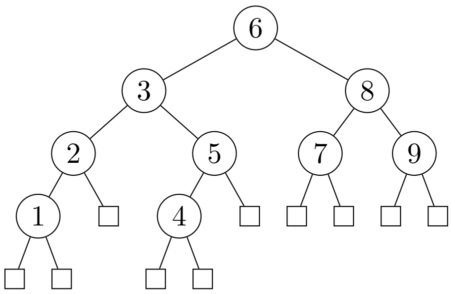
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
1, 3, 4, 6, 8		
2, 4, 6, 7, 9		
4, 5, 6		
2, 4, 6, 8		
4, 6		

Opgave 119 (4 %) check

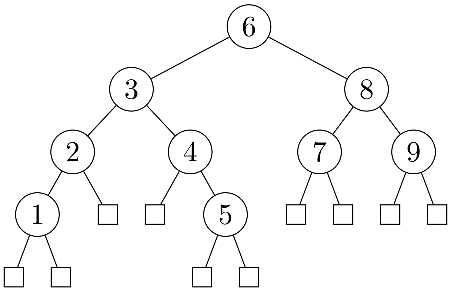
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



	Ja	Nej
1, 3, 4, 8		
1, 4		
1, 4, 6		
1, 3, 4, 7, 9		
1, 2, 4, 5, 7, 9		

Opgave 120 (4 %) check

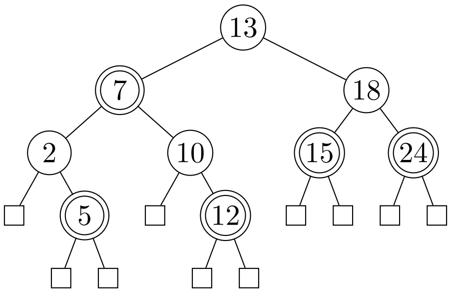
For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.



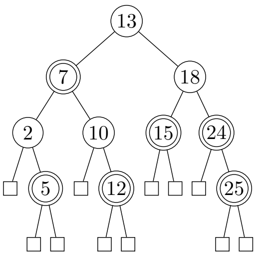
	Ja	Nej
1, 3, 5, 8		
1, 5, 6		
1, 5		
1, 3, 5, 7, 9		
1, 2, 4, 5, 8		

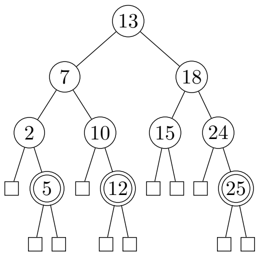
Indsættelse i rød-sort træer

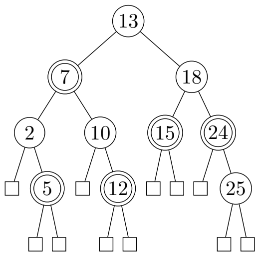
Opgave 121 (4 %) check

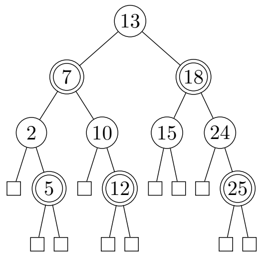


Angiv det resulterende rød-sorte træ når man indsætter 25 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

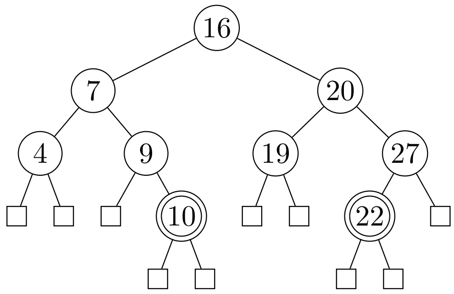




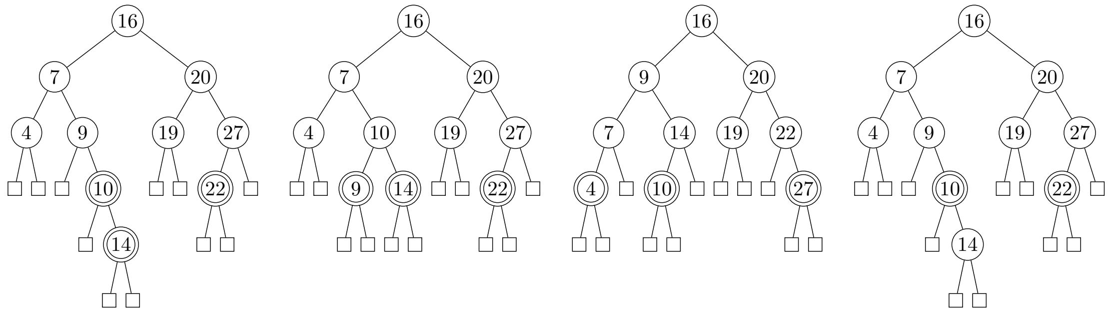




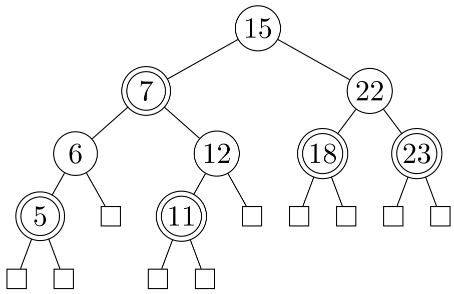
Opgave 122 (4 %) check



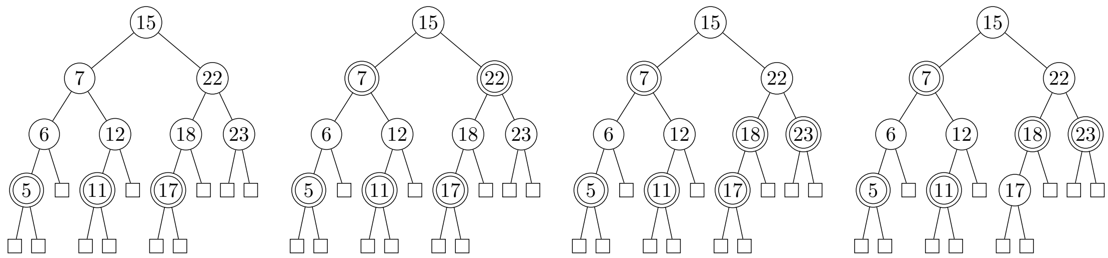
Angiv det resulterende rød-sorte træ når man indsætter 14 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).



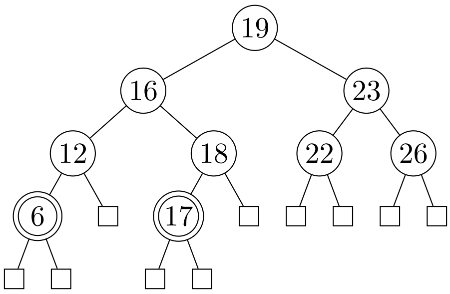
Opgave 123 (4 %) check



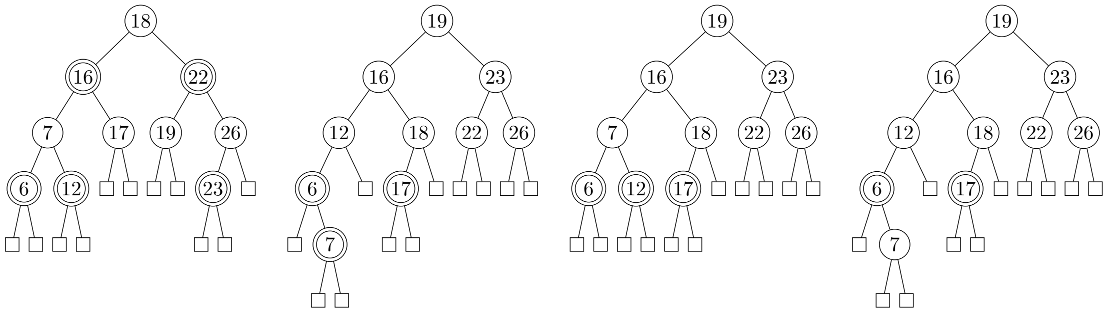
Angiv det resulterende rød-sorte træ når man indsætter 17 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).



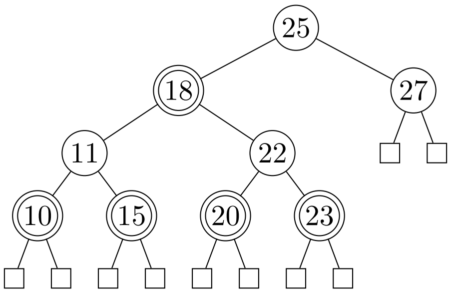
Opgave 124 (4 %) check



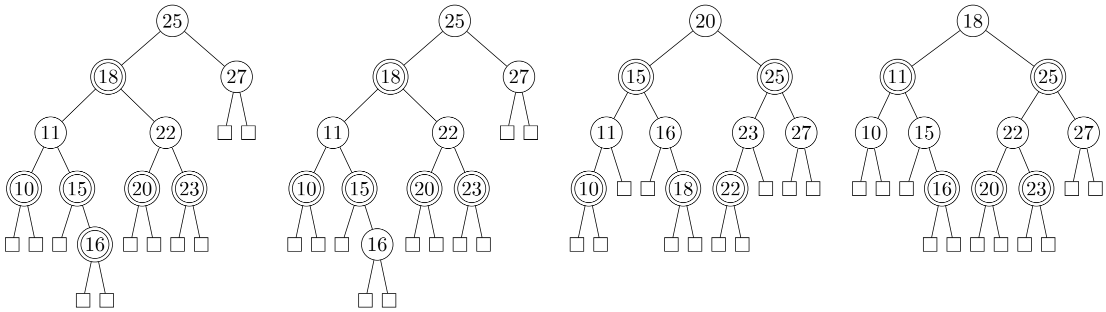
Angiv det resulterende rød-sorte træ når man indsætter 7 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).



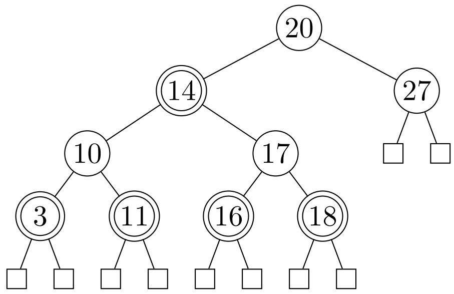
Opgave 125 (4 %) check



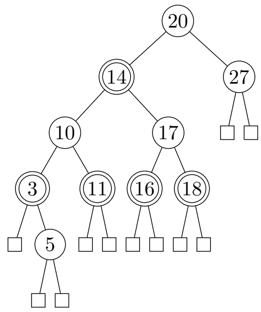
Angiv det resulterende rød-sorte træ når man indsætter 16 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

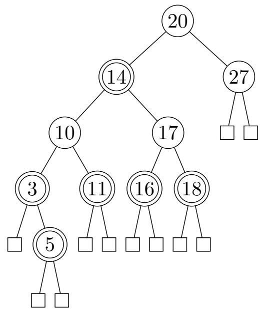


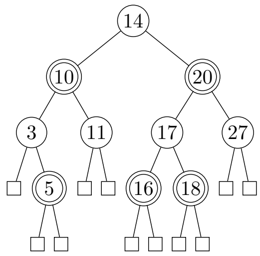
Opgave 126 (4 %) check

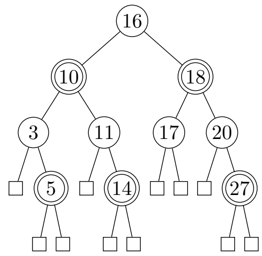


Angiv det resulterende rød-sorte træ når man indsætter 5 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

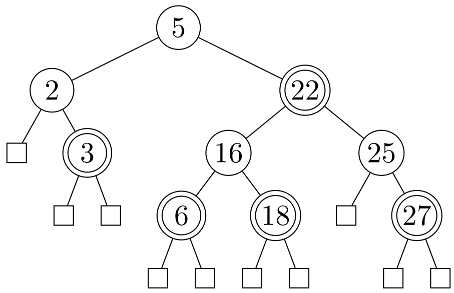




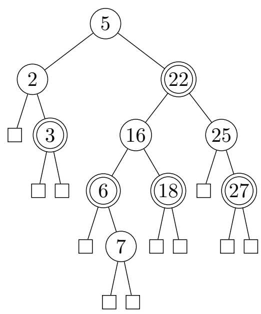


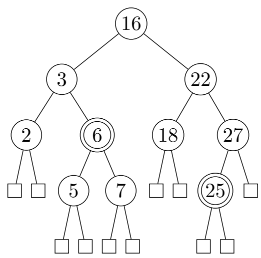


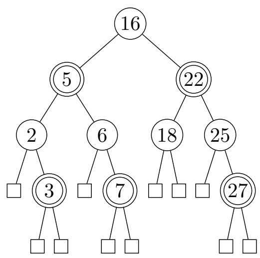
Opgave 127 (4 %) check

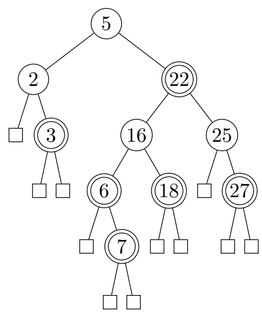


Angiv det resulterende rød-sorte træ når man indsætter 7 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

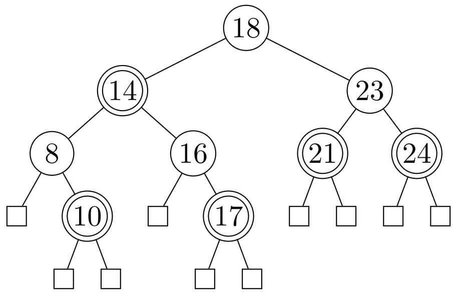




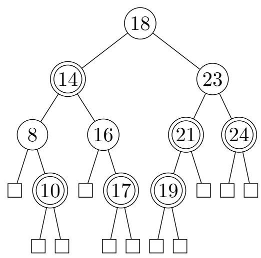
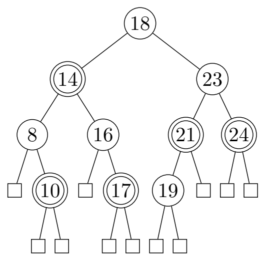
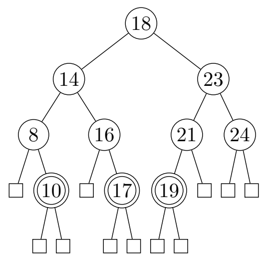
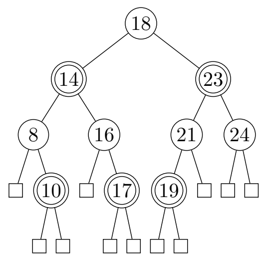




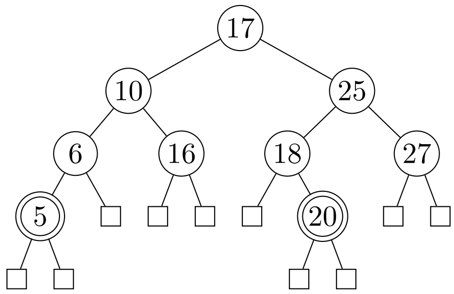
Opgave 128 (4 %) check



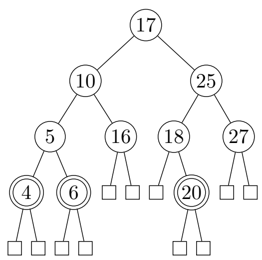
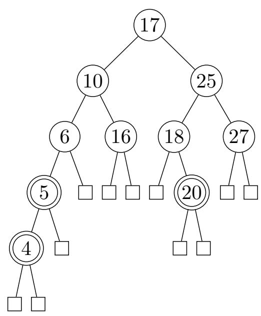
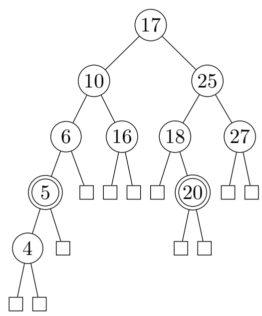
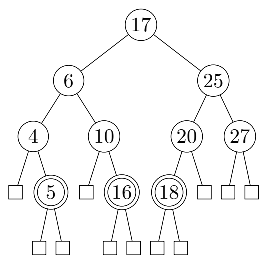
Angiv det resulterende rød-sorte træ når man indsætter 19 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

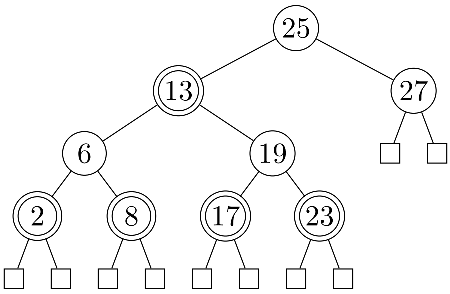
Opgave 129 (4 %) check



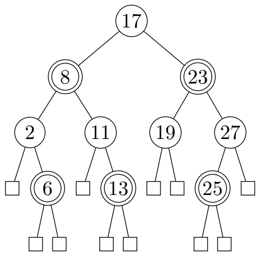
Angiv det resulterende rød-sorte træ når man indsætter 4 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

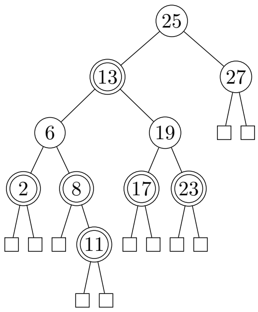
   

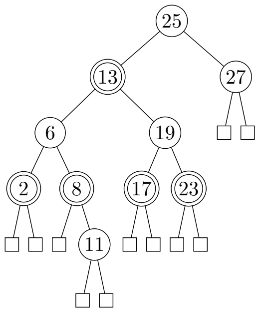
Opgave 130 (4 %) check

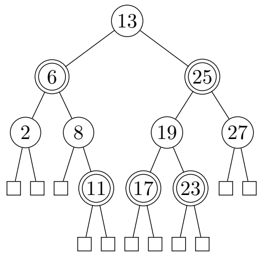


Angiv det resulterende rød-sorte træ når man indsætter 11 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).









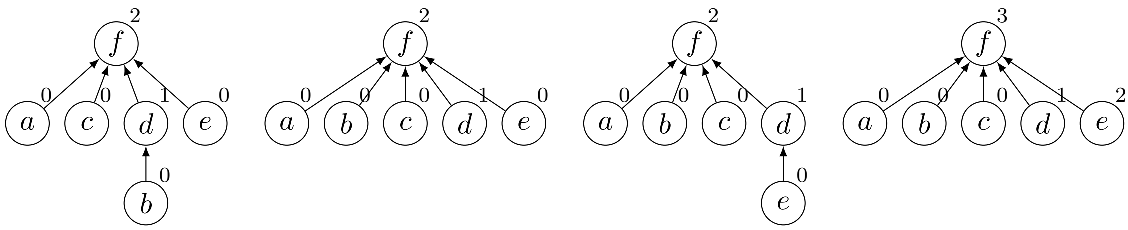
Union-find

Opgave 131 (4 %)

check

Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

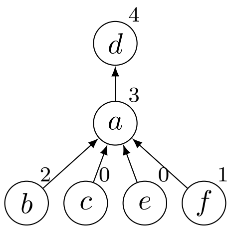
MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(b, d)
UNION(d, e)
UNION(c, f)
UNION(d, c)
UNION(a, e)
FIND-SET(b)

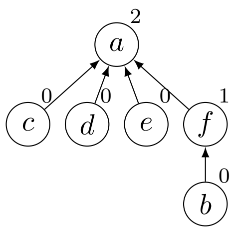


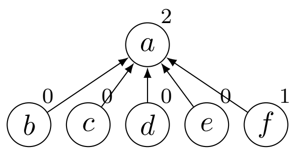
Opgave 132 (4 %) check

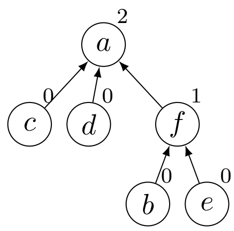
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(e, f)
UNION(f, b)
UNION(c, a)
UNION(b, c)
UNION(e, d)
FIND-SET(a)





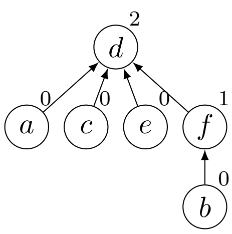


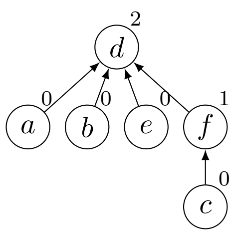


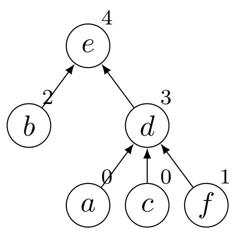
Opgave 133 (4 %) check

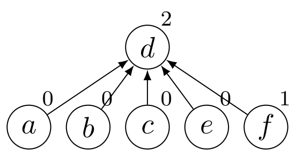
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(c, f)
UNION(f, b)
UNION(a, d)
UNION(f, a)
UNION(c, e)
FIND-SET(b)





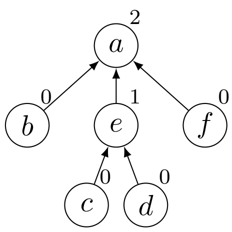


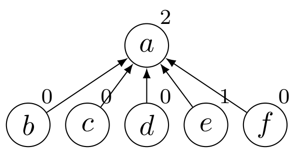


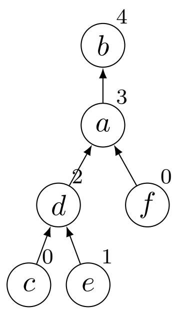
Opgave 134 (4 %) check

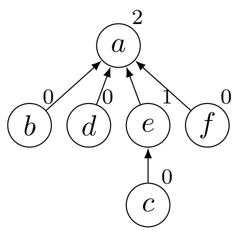
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(c, e)
UNION(e, d)
UNION(f, a)
UNION(c, f)
UNION(d, b)
FIND-SET(a)





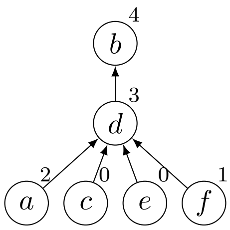


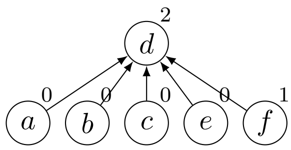


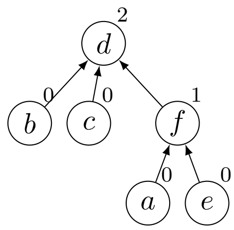
Opgave 135 (4 %) check

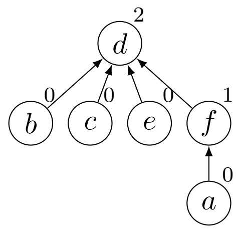
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(e, f)
UNION(f, a)
UNION(c, d)
UNION(a, c)
UNION(e, b)
FIND-SET(b)





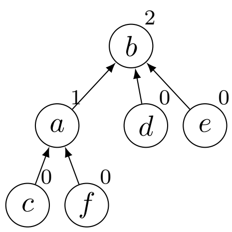


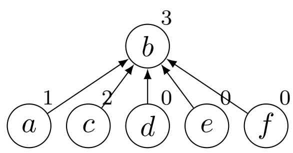


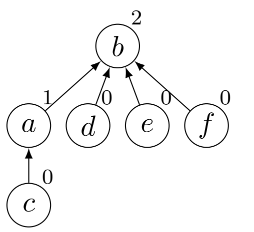
Opgave 136 (4 %) check

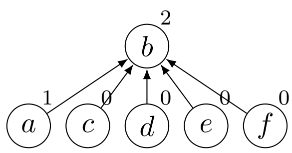
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(f, a)
UNION(a, c)
UNION(d, b)
UNION(c, b)
UNION(e, f)
FIND-SET(b)





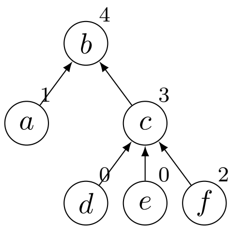


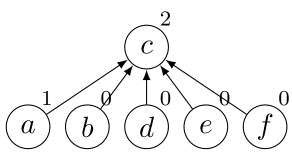


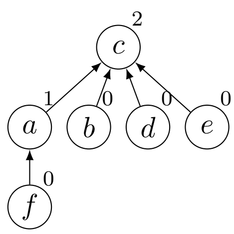
Opgave 137 (4 %) check

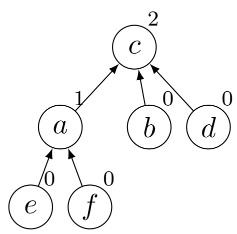
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(e, a)
UNION(a, f)
UNION(d, c)
UNION(f, c)
UNION(e, b)
FIND-SET(a)





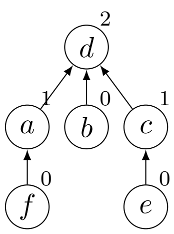


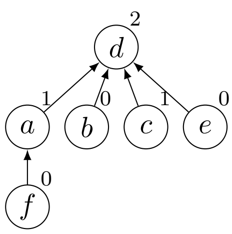


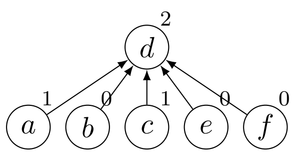
Opgave 138 (4 %) check

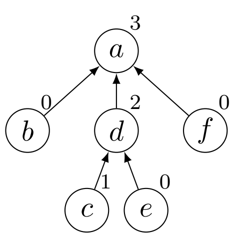
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(e, c)
UNION(b, d)
UNION(e, d)
UNION(f, a)
UNION(e, f)
FIND-SET(b)





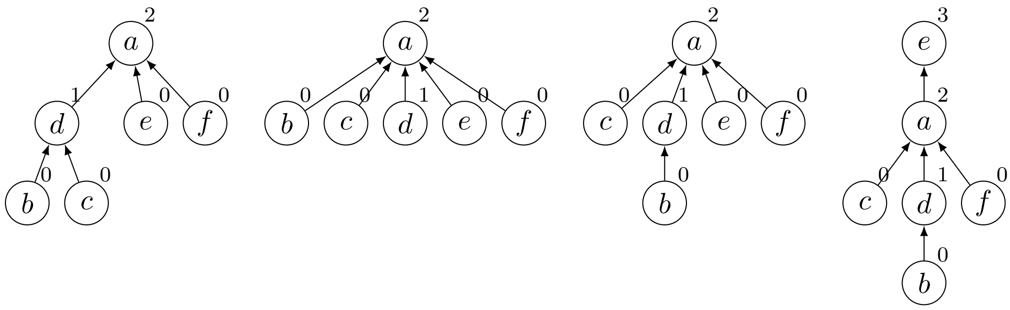




Opgave 139 (4 %) check

Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

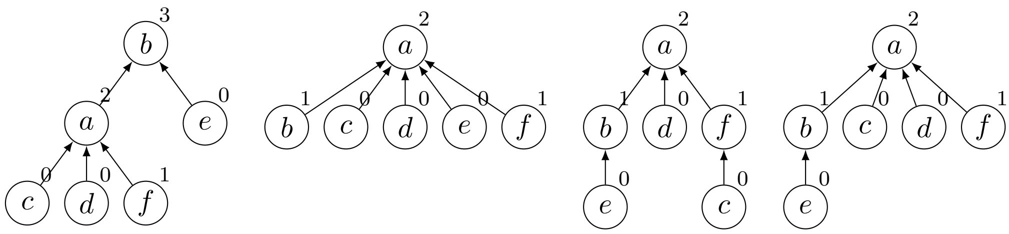
$\textsc{makeset}(a)$
$\textsc{makeset}(b)$
$\textsc{makeset}(c)$
$\textsc{makeset}(d)$
$\textsc{makeset}(e)$
$\textsc{makeset}(f)$
$\textsc{union}(b, d)$
$\textsc{union}(c, d)$
$\textsc{union}(f, a)$
$\textsc{union}(b, a)$
$\textsc{union}(c, e)$
$\textsc{Find-Set}(a)$



Opgave 140 (4 %) check

Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

$\textsc{makeset}(a)$
$\textsc{makeset}(b)$
$\textsc{makeset}(c)$
$\textsc{makeset}(d)$
$\textsc{makeset}(e)$
$\textsc{makeset}(f)$
$\textsc{union}(c,\, f)$
$\textsc{union}(d,\, a)$
$\textsc{union}(f,\, d)$
$\textsc{union}(e,\, b)$
$\textsc{union}(c,\, e)$
$\textsc{Find-Set}(b)$



Rekursionsligninger

Opgave 141 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

$\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$ $\Theta(n^2 \log n)$ $\Theta(n^3)$

$T(n) = 3 \cdot T(n/9) + 1$

$T(n) = 3 \cdot T(n/4) + n^2$

$T(n) = T(n-1) + \log n$

$T(n) = T(n-1) + 2$

$T(n) = 2 \cdot T(n/4) + 3$

Opgave 142 (4 %)

check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 9 \cdot T(n/3) + 1$							
$T(n) = T(n/2) + 5$							
$T(n) = 9 \cdot T(n/3) + n^2$							
$T(n) = 4 \cdot T(n/5) + n$							
$T(n) = 4 \cdot T(n/5) + n^3$							

Opgave 143 (4 %)

check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 2 \cdot T(n/4) + 1$							
$T(n) = T(n-1) + n^2$							
$T(n) = 3 \cdot T(n/9) + 3$							
$T(n) = 3 \cdot T(n/4) + n$							
$T(n) = 4 \cdot T(n/5) + n^2$							

Opgave 144 (4 %)

check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = T(n-1) + 1$							
$T(n) = 3 \cdot T(n/4) + n$							
$T(n) = 3 \cdot T(n/9) + 1$							
$T(n) = 4 \cdot T(n/4) + n$							
$T(n) = T(n/5) + 3$							

Opgave 145 (4 %)

check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 4 \cdot T(n/5) + n$							
$T(n) = 3 \cdot T(n/5) + n$							
$T(n) = 4 \cdot T(n/2) + n^2$							
$T(n) = T(n-1) + n^2$							
$T(n) = 3 \cdot T(n/9) + 3$							

Opgave 146 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 8 \cdot T(n/2) + 3$							
$T(n) = 4 \cdot T(n/5) + n$							
$T(n) = 5 \cdot T(n/5) + n$							
$T(n) = T(n-1) + \log n$							
$T(n) = 2 \cdot T(n/4) + 1$							

Opgave 147 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 3 \cdot T(n/5) + n$							
$T(n) = 9 \cdot T(n/3) + n^2$							
$T(n) = 2 \cdot T(n/4) + 3$							
$T(n) = T(n/3) + 2$							
$T(n) = 2 \cdot T(n/2) + n$							

Opgave 148 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = T(n-1) + 3$							
$T(n) = 9 \cdot T(n/3) + n^2$							
$T(n) = 4 \cdot T(n/2) + n^2$							
$T(n) = T(n-1) + \log n$							
$T(n) = T(n-1) + n^2$							

Opgave 149 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 4 \cdot T(n/2) + 1$							
$T(n) = 3 \cdot T(n/5) + n^3$							
$T(n) = 3 \cdot T(n/9) + 1$							
$T(n) = T(n/4) + 1$							
$T(n) = 9 \cdot T(n/3) + 3$							

Opgave 150 (4 %) check

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

$\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$ $\Theta(n^2 \log n)$ $\Theta(n^3)$

$T(n) = 4 \cdot T(n/2) + n^2$

$T(n) = T(n/2) + 1$

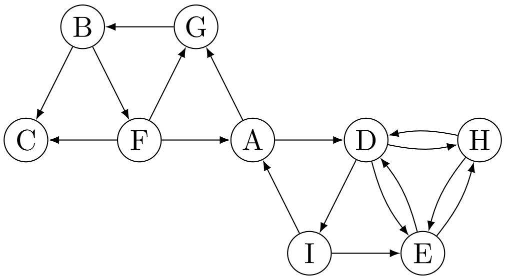
$T(n) = 2 \cdot T(n/4) + 3$

$T(n) = T(n-1) + n^2$

$T(n) = 3 \cdot T(n/5) + n^2$

BFS

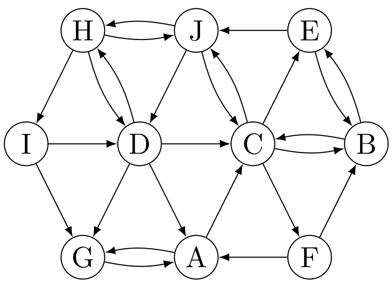
Opgave 151 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A D G E H I B F C   A G D B E H I C F   A D E H I G B C F   A D G E H I B C F

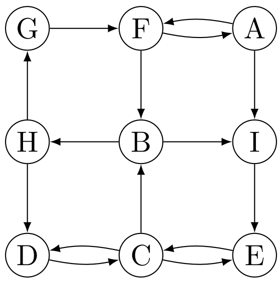
Opgave 152 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A G C F J B E D H I   A C G B E F J H D I   A C B E J D G H I F   A C G B E F J D H I

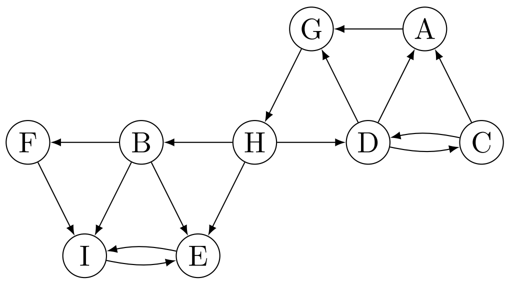
Opgave 153 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A I F E B C H D G   A F B H D C E G I   A F I B E H C G D   A F I B E H C D G

Opgave 154 (4 %) check

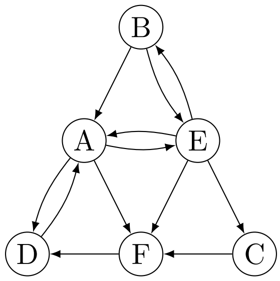


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A G H B E I F D C   A G H D B E C F I   A G H B D E I F C   A G H B D E F I C

Opgave 155 (4 %)

check

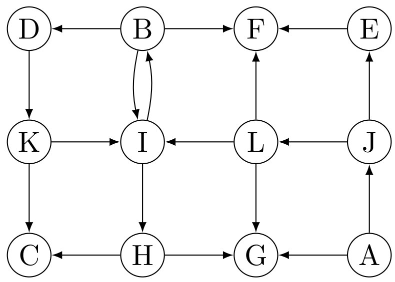


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A D E F B C   A F E D B C   A D E F C B   A D E B C F

Opgave 156 (4 %)

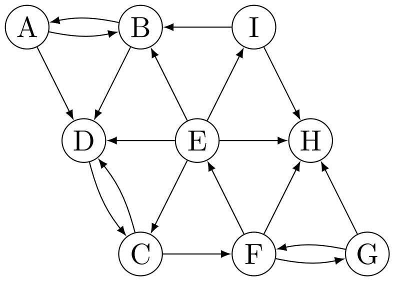
check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A G J E F L I B D K C H   A G J L E I F B H D C K   A G J E L F I H B C D K   A G J E L F I B H D C K

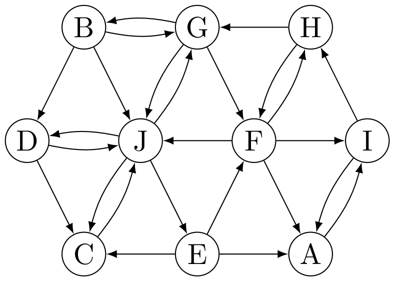
Opgave 157 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A D B C F E G H I A B D C F E H I G A B D C F H G E I A B D C F E G H I

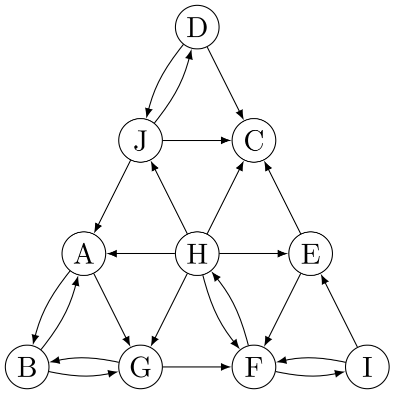
Opgave 158 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A I H F J C D E G B A I H F G J B E D C A I H G F B J D E C A I H F G J B C D E

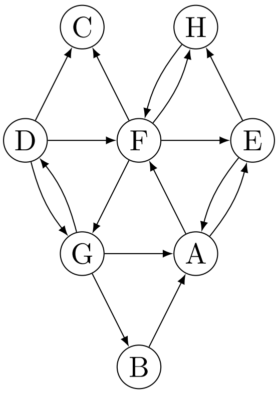
Opgave 159 (4 %) check



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A B G F H I C E J D   A B G F H I J E C D   A B G F H C E J D I   A B G F I H E C J D

Opgave 160 (4 %) check

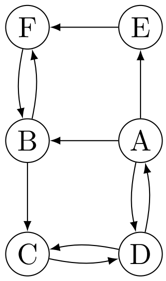


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A F E G H C D B   A E F H C G D B   A E H F C G B D   A E F H C G B D

Lovlige bredde først træer

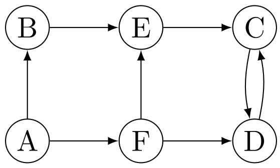
Opgave 161 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,B) (A,D) (A,E) (B,C) (E,F)		
(A,B) (A,D) (A,E) (B,C) (B,F)		
(A,D) (A,E) (D,C) (E,F) (F,B)		
(A,B) (A,E) (B,C) (B,F) (C,D)		
(A,B) (A,D) (A,E) (D,C) (E,F)		

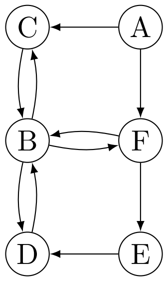
Opgave 162 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,B) (A,F) (B,E) (D,C) (F,D)		
(A,B) (A,F) (E,C) (F,D) (F,E)		
(A,B) (A,F) (B,E) (E,C) (F,D)		
(A,B) (A,F) (B,E) (C,D) (E,C)		
(A,B) (A,F) (C,D) (E,C) (F,E)		

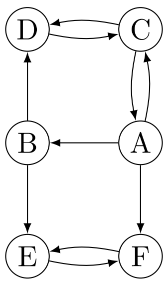
Opgave 163 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,C) (A,F) (E,D) (F,B) (F,E)		
(A,C) (B,F) (C,B) (E,D) (F,E)		
(A,C) (A,F) (B,D) (C,B) (F,E)		
(A,C) (A,F) (D,B) (E,D) (F,E)		
(A,C) (A,F) (B,D) (F,B) (F,E)		

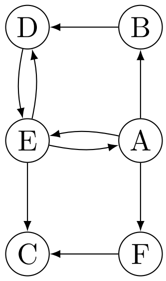
Opgave 164 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,B) (A,F) (B,D) (B,E) (D,C)		
(A,B) (A,C) (A,F) (B,D) (F,E)		
(A,B) (A,C) (B,E) (C,D) (E,F)		
(A,B) (A,C) (A,F) (C,D) (F,E)		
(A,B) (A,C) (A,F) (B,E) (C,D)		

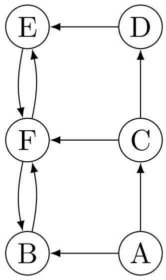
Opgave 165 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,B) (A,E) (A,F) (E,C) (E,D)		
(A,B) (A,E) (A,F) (E,D) (F,C)		
(A,B) (A,F) (B,D) (D,E) (F,C)		
(A,B) (A,E) (A,F) (B,D) (E,C)		
(A,B) (A,E) (A,F) (B,D) (F,C)		

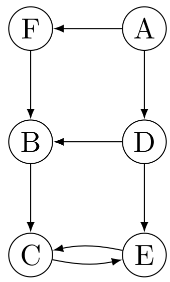
Opgave 166 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,B) (A,C) (B,F) (C,D) (F,E)		
(A,B) (A,C) (C,D) (C,F) (D,E)		
(A,B) (A,C) (C,D) (D,E) (E,F)		
(A,B) (A,C) (C,D) (C,F) (F,E)		
(A,C) (C,D) (C,F) (F,B) (F,E)		

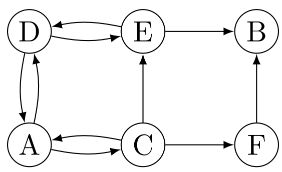
Opgave 167 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,D) (A,F) (B,C) (D,E) (F,B)		
(A,D) (A,F) (D,E) (E,C) (F,B)		
(A,D) (A,F) (B,C) (C,E) (D,B)		
(A,D) (A,F) (D,B) (D,E) (E,C)		
(A,D) (A,F) (B,C) (D,B) (D,E)		

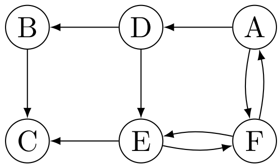
Opgave 168 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,C) (A,D) (C,E) (C,F) (E,B)		
(A,C) (A,D) (C,E) (C,F) (F,B)		
(A,C) (A,D) (C,F) (D,E) (E,B)		
(A,C) (C,E) (C,F) (E,B) (E,D)		
(A,C) (A,D) (C,F) (D,E) (F,B)		

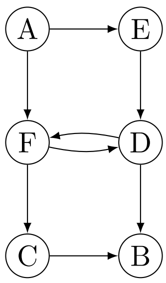
Opgave 169 (4 %) check



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,D) (A,F) (B,C) (D,B) (F,E)		
(A,D) (A,F) (B,C) (D,B) (D,E)		
(A,D) (D,B) (D,E) (E,C) (E,F)		
(A,D) (A,F) (D,B) (D,E) (E,C)		
(A,D) (A,F) (D,B) (E,C) (F,E)		

Opgave 170 (4 %) check

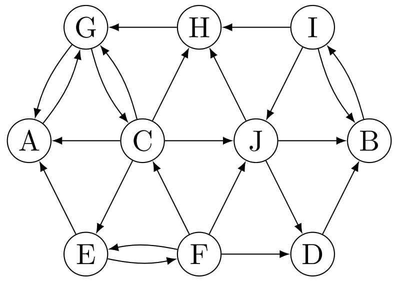


Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

	Ja	Nej
(A,E) (A,F) (D,B) (E,D) (F,C)		
(A,E) (D,B) (D,F) (E,D) (F,C)		
(A,E) (C,B) (D,F) (E,D) (F,C)		
(A,E) (A,F) (D,B) (F,C) (F,D)		
(A,E) (A,F) (C,B) (F,C) (F,D)		

DFS

Opgave 171 (4 %) check



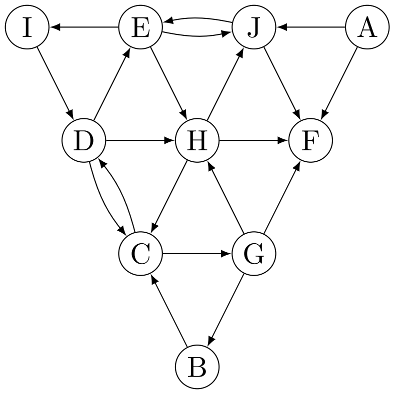
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

H J I B D F E C G A   J H I B D F E C G A   I D B F J H E C G A   H I B D J F E C G A

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(C, H)				
(J, H)				
(A, G)				
(J, D)				

Opgave 172 (4 %) check



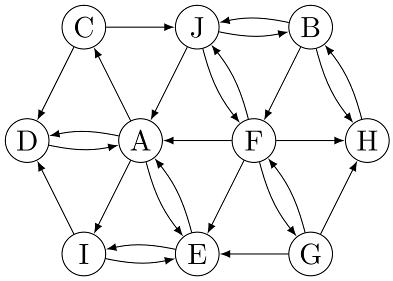
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

F B G D C H I E J A   B G D C I H E J F A   F D B G C H I E J A   I B G D C H E J F A

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(E, I)				
(E, J)				
(J, F)				
(A, F)				

Opgave 173 (4 %) check



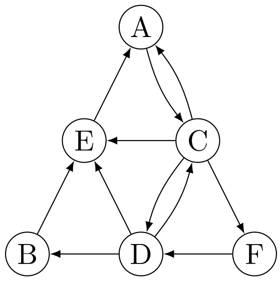
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **discovery time**.

A C J F E I D G H B   A C J B F G H E I D   A C D J B F E I G H   A C D E I J B F H G

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(J, B)				
(A, I)				
(B, J)				
(I, D)				

Opgave 174 (4 %) check



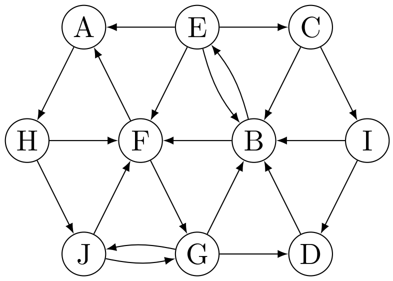
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **discovery time**.

A C F D E B   A C D E F B   A C F D B E   A C D B E F

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(F, D)				
(C, E)				
(B, E)				
(E, A)				

Opgave 175 (4 %) check



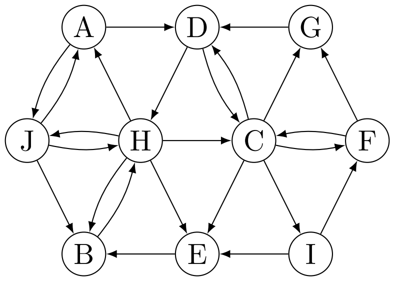
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **discovery time**.

A H F G B E C I D J   A H F J G B D E C I   A H F G J B E C I D   A H J G B F E C I D

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(C, B)				
(G, D)				
(G, J)				
(A, H)				

Opgave 176 (4 %) check



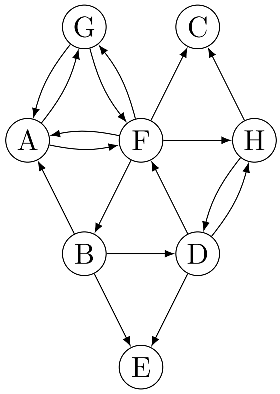
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

I G F J H B E C D A   J H B E G F I C D A   I G F E B H C J D A   D G F B E I C H J A

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(C, D)				
(A, D)				
(I, E)				
(D, H)				

Opgave 177 (4 %) check



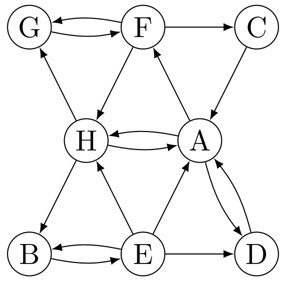
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **discovery time**.

A F G B C H D E   A F G H D E C B   A F B D E H C G   A F G B D H C E

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(G, F)				
(A, G)				
(F, B)				
(F, G)				

Opgave 178 (4 %) check



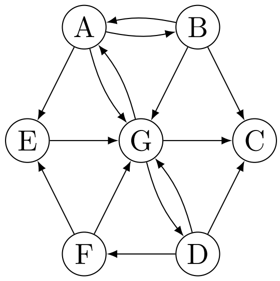
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

E B G C H F D A   E B H G C F D A   D C G E B H F A   D E B C F G H A

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(D, A)				
(A, H)				
(B, E)				
(E, D)				

Opgave 179 (4 %) check



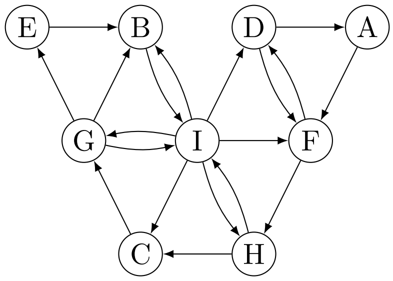
Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

F D C G E B A   E F C D G B A   C E F D G B A   E F D G C B A

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(D, F)				
(D, G)				
(A, G)				
(G, C)				

Opgave 180 (4 %) check



Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **discovery time**.

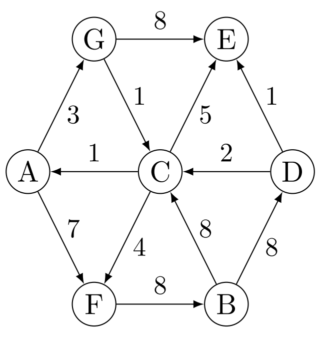
A F H C G E B I D   A F D H C G I B E   A F D H C G B I E   A F D H C I G B E

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(I, C)				
(B, I)				
(E, B)				
(H, I)				

Dijkstras algoritme

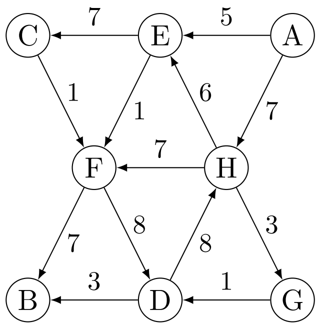
Opgave 181 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A G C F B D E   A F G B C E D   A F B C E D G   A G C F E B D

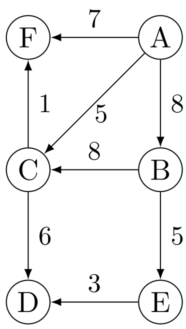
Opgave 182 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A E F H G D B C A E H C F G B D A E C F B D H G A E F H G D C B

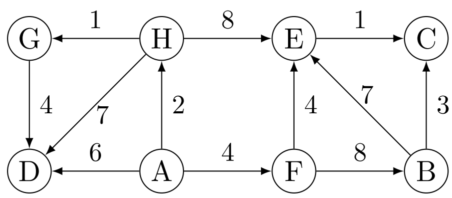
Opgave 183 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A C F B E D A C F B D E A B C D F E A B C F E D

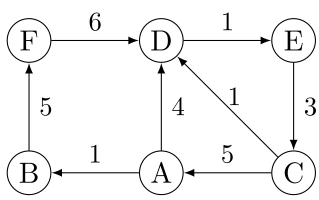
Opgave 184 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A H G F D E C B   A H G F E C D B   A D F B C E H G   A D F H B E G C

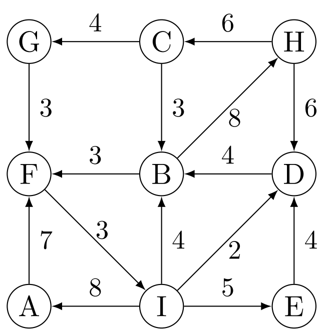
Opgave 185 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A B D F E C   A B D E F C   A B D E C F   A B F D E C

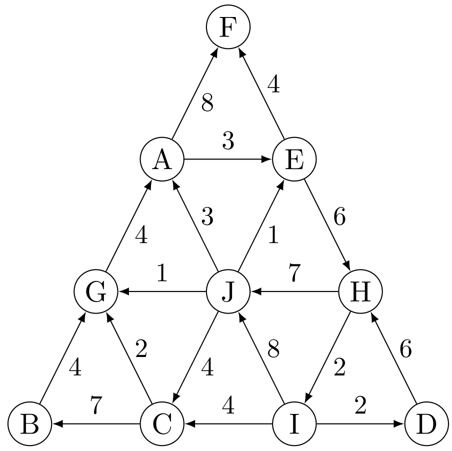
Opgave 186 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A F I D B E H C G   A F I D B H C G E   A F I B D E H C G   A F I B H C G D E

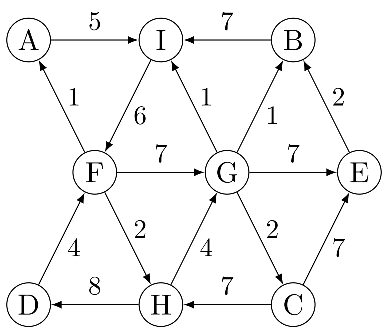
Opgave 187 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A E F H I C B G D J   A E F H I J C D G B   A E F H I D C J G B   A E F H I D C G B J

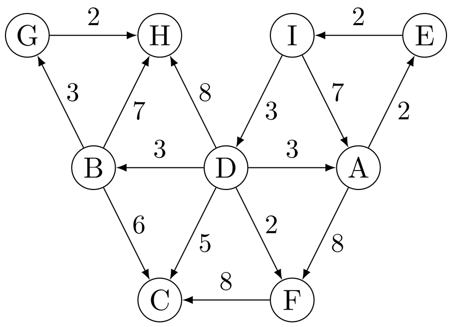
Opgave 188 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A I F H G B C E D   A I F G B C E H D   A I F H G B C D E   A I F G H B C E D

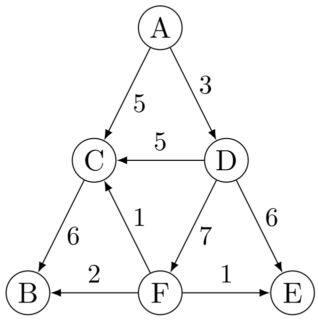
Opgave 189 (4 %) check



Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A E I D F C B G H   A E I D B C G H F   A E F I C D B H G   A E I D F B C G H

Opgave 190 (4 %) check

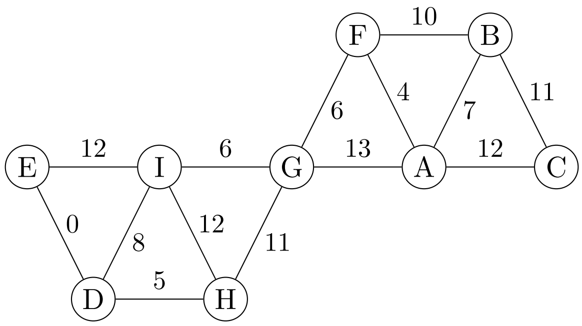


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A C D B E F   A D C B E F   A C B D E F   A D C E F B

Prims algoritme

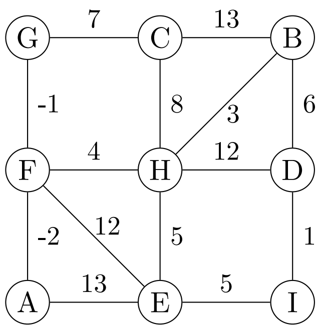
Opgave 191 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A F B G C I H D E   A F B G I C D E H   A F G I D E H B C   A F G I B D E H C

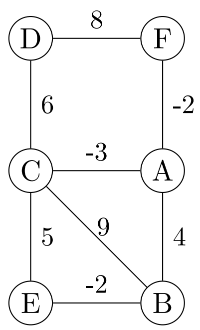
Opgave 192 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A F G H C B E D I   A F G H C B E I D   A F G C H B D I E   A F G H B E I D C

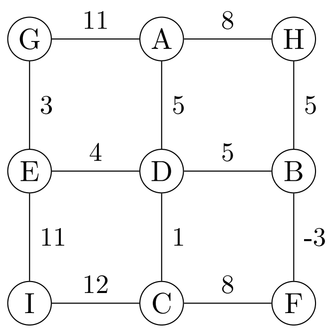
Opgave 193 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A C F E B D   A C E B D F   A C F D B E   A C F B E D

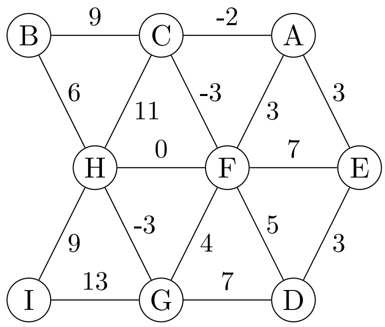
Opgave 194 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A D C F B H I E G   A D C E B F G H I   A D C H E B F G I   A D C E G B F H I

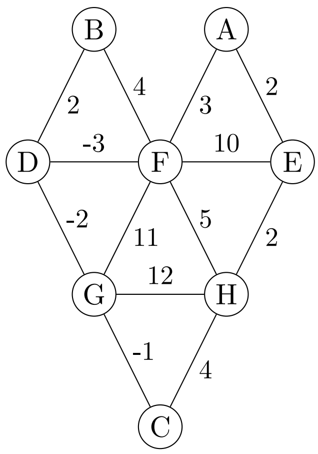
Opgave 195 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A C F H G B E I D A C F H G D B E I A C F H G E D B I A C F H G D E I B

Opgave 196 (4 %) check

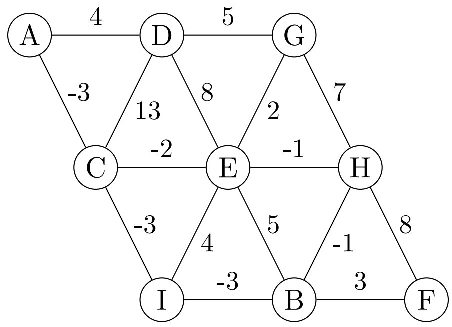


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A E F D G C H B A E H F D G C B A E F D G C B H A E H C G D F B

Opgave 197 (4 %)

check

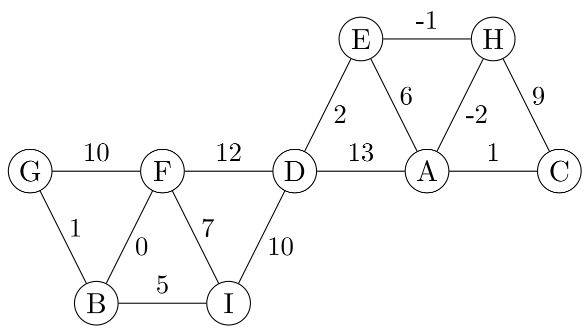


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A C I B H F E G D   A C I B H E G F D   A C I B E H G F D   A C I B H E G D F

Opgave 198 (4 %)

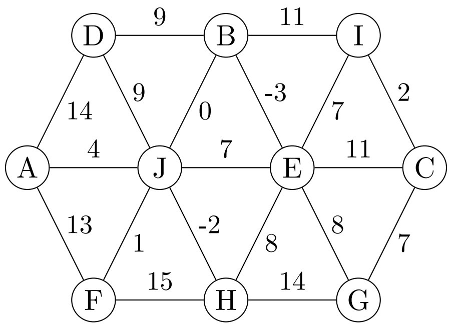
check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A H E C D I B F G   A H E D I B F G C   A H E D C I B F G   A H E D C I F B G

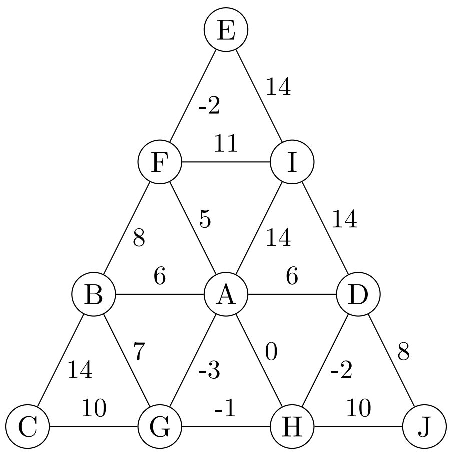
Opgave 199 (4 %) check



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A J H B E F I C D G   A J H B E F I G C D   A J H E B D I C G F   A J H B E F I C G D

Opgave 200 (4 %) check



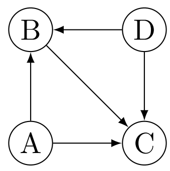
Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A G H D F E B J C I   A G H D J F E B C I   A G H D J I F E B C   A G H D J B F E C I

Topologisk sortering

Opgave 201 (4 %)

check

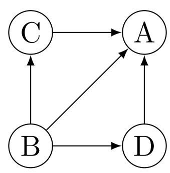


Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A D B C		
D A B C		
C A B D		
C D B A		
A B D C		

Opgave 202 (4 %)

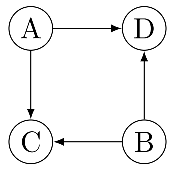
check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A C D B		
D C B A		
B D C A		
C D B A		
B C D A		

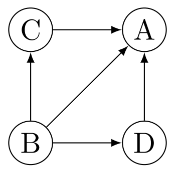
Opgave 203 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A B D C		
C B A D		
B A C D		
D B C A		
B A D C		

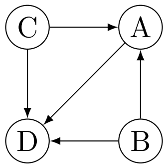
Opgave 204 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
B A C D		
B D C A		
B C D A		
A D C B		
C B D A		

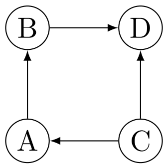
Opgave 205 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
B D A C		
C B A D		
D B A C		
B C D A		
B C A D		

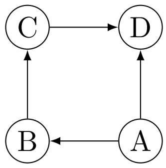
Opgave 206 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
C D B A		
C A D B		
B A C D		
C B A D		
C A B D		

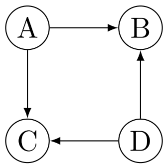
Opgave 207 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
C B A D		
A C B D		
A B C D		
A B D C		
B A C D		

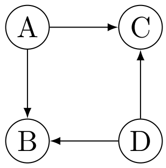
Opgave 208 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A D C B		
D B A C		
D A C B		
A D B C		
D C B A		

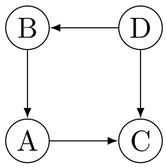
Opgave 209 (4 %) check



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A D C B		
D A C B		
C A B D		
D A B C		
C A D B		

Opgave 210 (4 %) check



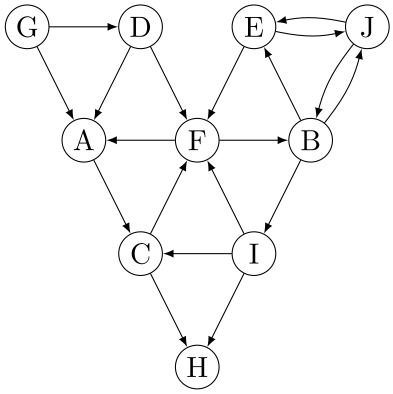
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
D B C A		
D A B C		
A B D C		
D B A C		
B D A C		

Stærke sammenhængskomponenter

Opgave 211 (4 %)

check

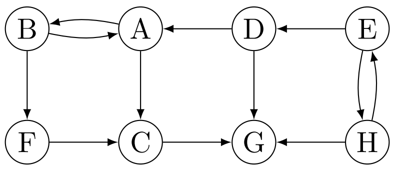


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9 10

Opgave 212 (4 %)

check

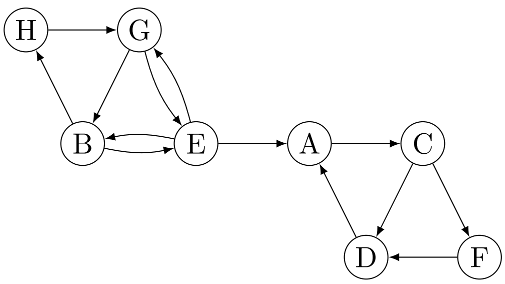


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8

Opgave 213 (4 %)

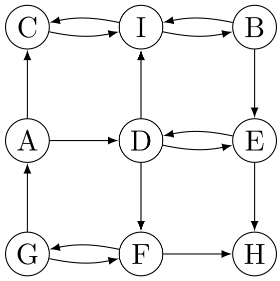
check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8

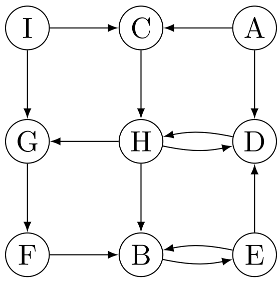
Opgave 214 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

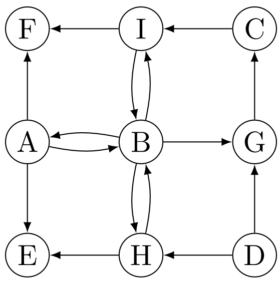
Opgave 215 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

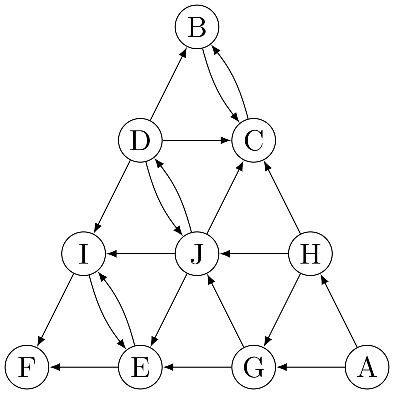
Opgave 216 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

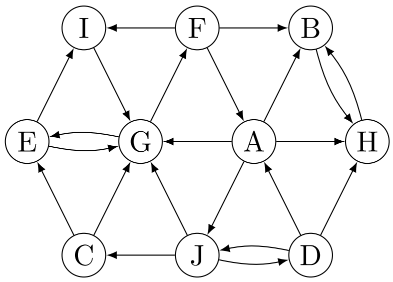
Opgave 217 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9 10

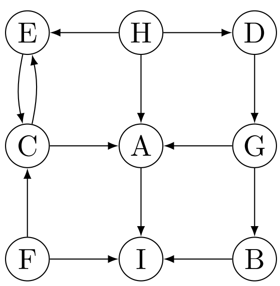
Opgave 218 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9 10

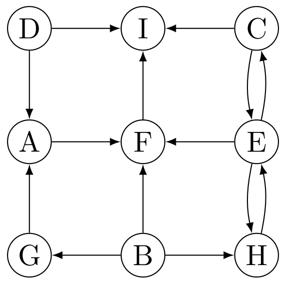
Opgave 219 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Opgave 220 (4 %) check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Amortiseret analyse

Opgave 221 check

Antag en liste L bruges til at opbevare en mængde af forskellige tal. Følgende to operationer ønskes understøttet: Add(x) tilføjer x til listen (x antages at være forskellig fra alle tal i listen), og RemoveLargerHalf fjerner og returnerer de $\lceil |L|/2 \rceil$ største elementer fra listen. Add(x) tilføjer blot det nye element bagerst i listen L i worst-case $O(1)$ tid. RemoveLargerHalf anvender først deterministisk selektion til at finde det $\lfloor |L|/2 \rfloor + 1$ mindste element e i L i worst-case $O(|L|)$ tid, hvorefter L løbes igennem og alle elementer fjernes der er større end eller lig e.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at begge operationer tager amortiseret $O(1)$ tid.

	Ja	Nej
$\lvert L\rvert \cdot \log \lvert L\rvert$		
$2\lvert L\rvert$		
$\log \lvert L\rvert$		
$\lvert L\rvert$		
$\lvert L\rvert^2$		
$\lvert L\rvert/2$		

Opgave 222 check

Antag at et array X af størrelse n indeholder to stakke S og T, henholdsvis af størrelse s og t, således at $X[1..s] = S$ og $X[n+1-t..n] = T$, hvor toppen af de to stakke er henholdsvis $X[s]$ og $X[n+1-t]$. Når X bliver fuld, dvs. $s+t=n$, allokeres et dobbelt så stort array til X, og S og T kopieres over i dette array.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at stakoperationerne PUSH og POP på de to stakke tager amortiseret $O(1)$ tid.

	Ja	Nej
$\max\{0, 2(s+t)-n\}$		
$\max\{0, 2s+3t-n\}$		
$s+t$		
$s+n-t$		
$n-s-t$		
$t-s$		

Opgave 223 check

Givet en sorteret liste L med $N = 2^k - 1$ elementer, for et positivt heltal k, så kan man i $O(N)$ tid konstruere et perfekt balanceret binært søgetræ indeholdende L. I det følgende antages at vi kun laver sletninger i søgetræet, som beskrevet i [CLRS, kapitel 12.3], dvs. sletninger forsøger ikke at holde træet balanceret. Lad n betegne det aktuelle antal elementer i træet. For at holde træets højde logaritmisk i n, genopbygges træet som et perfekt balanceret binært søgetræ i $O(n)$ tid når halvdelen af elementerne er blevet slettet, dvs. når $n < N/2$, hvor N betegner antallet af elementer i træet sidste gang det blev genopbygget.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at slettelser tager amortiseret $O(\log n)$ tid.

	Ja	Nej
$(N-n) \cdot \log N$		
n		
N		
$\log n$		
$(N-n) \cdot \log n$		
$N-n$		

Opgave 224 check

Betragt en liste $L = (x_1, \ldots, x_N)$ af N heltal, hvorpå vi kan udføre følgende to operationer: $\text{Append}(x)$ tilføjer heltallet x til sidst i listen, og AddPairs erstatter for alle $i = 1..\lfloor |L|/2 \rfloor$ det $2i-1$'te og $2i$'te tal med deres sum, således at den nye liste har længde $\lceil |L|/2 \rceil$. F.eks. resulterer AddPairs på listen $3, 5, 7, 4, 11, 2, 6$ i den nye liste $8, 11, 13, 6$. Worst-case tiden for Append og AddPairs er henholdsvis $O(1)$ og $O(|L|)$.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at begge operationer tager amortiseret $O(1)$ tid.

	Ja	Nej
$N \cdot \log N$		
$N + \log N$		
$N/2$		
$\sqrt{N}$		
$\log N$		
N		

Opgave 225 check

Betragt en uordnet liste af n heltal, hvor vi i $O(1)$ tid kan indsætte et nyt heltal, og i $O(n)$ tid kan udføre operationen NegateRemove, som fjerner alle ikke-positive tal ($x \leq 0$) fra listen og erstatter ethvert positivt heltal $y > 0$ med det tilsvarende negative heltal $-y$. $\text{NegateRemove}(3, -4, -2, 7, 6, -2) = (-3, -7, -6)$.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at både indsættelser og NegateRemove tager amortiseret $O(1)$ tid, hvor P er antal positive tal i listen og N er antal ikke-positive tal i listen.

	Ja	Nej
P		
$N + P$		
N		
$N + 2P$		
$2N + P$		
$P + N/2$		

Opgave 226 check

Betragt et rød-sort søgetræ udvidet med operation INSERTCUT(x), som indsætter elementet x i søgetræet og sletter alle elementer der er mindre end x fra søgetræet. Hvis indsættelsen sletter k elementer, så tager dette worst-case $O((k+1)\log n)$ tid, hvor n er antal elementer i træet før INSERTCUT udføres.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at INSERTCUT tager amortiseret $O(\log n)$.

	Ja	Nej
n		
$n \cdot \log n$		
$\log n$		
$k \cdot \log n$		
$\sum_{i=1}^{n} \log i$		

Opgave 227 check

Antag vi ønsker at gemme en mængde af n tal vha. hashing med linear probing i et array af størrelse N. Vi garanterer at arrayet altid er mellem 1/4 og 3/4 fyldt. Hvis der bliver færre end $N/4$ eller flere end $3N/4$ tal i mængden genindsætter vi alle tal i et nyt array af størrelse $2n$, dvs. det nye array er 1/2 fyldt.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at det totalle antal genindsættelser i hashtabellerne er amortiseret $O(1)$ per indsættelse og slettelse i mængden.

	Ja	Nej
N		
$N/2 - \min(n, N-n)$		
$\min(n, N-n)$		
$\|2n - N\|$		
$N - n$		
$4\|N/2 - n\|$		

Opgave 228 check

En binær max-heap understøtter INSERT og HEAP-EXTRACT-MAX på en max-heap med n elementer i worst-case $O(\log n)$ tid.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at INSERT tager amortiseret $O(\log n)$ tid og HEAP-EXTRACT-MAX tager amortiseret $O(1)$ tid.

	Ja	Nej
$n \cdot \log n$		
$(\log n)^2$		
n		
$\sum_{i=1}^{n} \log i$		
$\log n$		

Opgave 229 check

Betragt en binær max-heap implementeret i et array. Overløb håndteres ved at allokere et nyt array af dobbelt størrelse og kopiere indholdet af det gamle array til det nye array. Lad den aktuelle størrelse af arrayet være N og antallet af elementer i heapen n.

Angiv for hver af nedenstående funktioner om de er en potentialefunktion, hvormed man kan argumentere for at INSERT og HEAP-EXTRACT-MAX kræver amortiseret $O(\log n)$ tid.

	Ja	Nej
$N - 2n$		
$\log(N/n)$		
$\|2n - N\|$		
$\max(0, 2n - N)$		
$\log n$		
n		

Opgave 230 check

Betragt en stak implementeret i et array, hvor overløb håndteres ved at allokere et nyt større array og kopiere indholdet af det gamle array til det nye array. Angiv den amortiserede tid for en PUSH operation, når det nye array har nedenstående størrelse N og n er antal elementer på stakken før PUSH operationen.

	$O(1)$	$O(\log n)$	$O(\sqrt{n})$	$O(n)$
$N = n + \lceil \sqrt{n} \rceil$				
$N = 3n$				
$N = n + 1$				
$N = \lceil \frac{3}{2} n \rceil$				
$N = 2n$				

Invarianter

Opgave 231 (4 %) check

Givet to ikke-negative heltal n og m, så beregner nedenstående algoritme $n \cdot m$.

Algoritme Multiplikation(n)
Inputbetingelse : Heltal $n \geq 0$ og $m \geq 0$
Outputkrav : $r = n_0 \cdot m_0$
Metode : $r \leftarrow 0$
$\{I\}$ **while** $n > 0$ **do**
  if n er ulige **then**
    $r \leftarrow r + m$
    $n \leftarrow n - 1$
  else
    $m \leftarrow m * 2$
    $n \leftarrow n/2$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Multiplikation, hvor n_0 og m_0 angiver værdierne for henholdsvis n og m i starten.

	Ja	Nej
$n_0 \cdot m_0 + r = n \cdot m$		
$n_0 \cdot m_0 = r + n \cdot m$		
$0 \leq m \leq m_0$		
$0 \leq n \leq n_0$		
$r = m \cdot n$		

Opgave 232 (4 %) check

Givet et positive heltal x og y, så beregner nedenstående algoritme x^y.

Algoritme Power(x, y)
Inputbetingelse : Heltal $x \geq 1$ og $y \geq 1$
Outputkrav : $r = x^y$
Metode : $r \leftarrow 1$
  $\{I\}$ **while** $y \geq 1$ **do**
    if y ulige **then**
      $y \leftarrow y - 1$
      $r \leftarrow r * x$
    else
      $y \leftarrow y/2$
      $x \leftarrow x * x$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Power, hvor x_0 og y_0 angiver start værdierne for x og y.

	Ja	Nej
$r = x_0^{y_0 - y}$		
$x^y = r \cdot x_0^{y_0}$		
$r = x_0^{y_0}$		
$r = x^y$		
$x_0^{y_0} = r \cdot x^y$		

Opgave 233 (4 %) check

Givet et positivt heltal n, så beregner nedenstående algoritme n^2.

Algoritme Square(n)
Inputbetingelse : Heltal $n \geq 1$
Outputkrav : $r = n^2$
Metode : $r \leftarrow 0$
  $i \leftarrow 0$
  $\{I\}$ **while** $i < n$ **do**
    $i \leftarrow i + 1$
    $r \leftarrow r + i$
  $r \leftarrow 2 * r - n$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Square.

	Ja	Nej
$r \geq i \geq 0$		
$i \geq 0 \ \wedge \ r = i^2$		
$0 \leq r \leq n$		
$0 \leq i \leq n$		
$i \geq 0 \ \wedge \ r = i(i+1)/2$		

Opgave 234 (4 %) check

Givet et array $A[1..n]$ indeholdende $n \geq 1$ heltal og et heltal x, så beregner nedenstående algoritme antallet af forekomster af x i A, dvs. $count(x, A) = |\{i \mid 1 \leq i \leq n \ \wedge \ A[i] = x\}|$.

Algoritme Count(x, A)
Inputbetingelse : Heltal x og array $A[1..n]$ med n heltal
Outputkrav : $r = count(x, A)$
Metode : $i \leftarrow 0$
$r \leftarrow 0$
$\{I\}$ **while** $i < n$ **do**
  $i \leftarrow i + 1$
  if $x = A[i]$ **then**
    $r \leftarrow r + 1$

For hvert af følgende udsagn, angiv om de er en invariant I for ovenstående algoritme Count.

	Ja	Nej
$i \leq n$		
$r = count(x, A[1..i + 1])$		
$r = count(x, A[1..i])$		
$r = 0$		
$i < n$		

Opgave 235 (4 %) check

Antag at et sorteret array $A[1..n]$ indeholder $n \geq 2$ forskellige heltal, dvs. $A[1] < A[2] < \cdots < A[n-1] < A[n]$. Givet et positivt heltal $x > 0$, identificerer nedenstående algoritme om der findes $1 \leq i < j \leq n$, hvor $x = A[j] - A[i]$.

Algoritme FindDiff$(A[1..n], x)$
Inputbetingelse : Sorteret array $A[1..n]$ med n forskellige heltal, heltal $x > 0$
Outputkrav : $1 \leq i < j \leq n$, hvor $x = A[j] - A[i]$;
ellers $j > n$ hvis intet sådant par findes
Metode : $i \leftarrow 1$;
$j \leftarrow 1$;
$\{I\}$ **while** $j \leq n$ **and** $x \neq A[j] - A[i]$ **do**
  if $x > A[j] - A[i]$ **then**
    $j \leftarrow j + 1$
  else
    $i \leftarrow i + 1$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen FindDiff.

	Ja	Nej
for alle $1 \leq i' < j' \leq n : (i' < i \ \vee \ j' < j) \Rightarrow x \neq A[j'] - A[i']$		
$1 \leq i \leq j \leq n$		
$i \leq j$		
for alle $1 \leq i' < j' \leq n : (i' \geq i \ \vee \ j' \geq j) \Rightarrow x \neq A[j'] - A[i']$		
$x \neq A[j] - A[i]$		

Opgave 236 (4 %) check

Lad $\|x\|$ betegne antal bit med værdien 1 i den binære repræsentation af et ikke-negativt heltal x, f.eks. $\|14_{10}\| = \|1110_2\| = 3$. Nedenstående algoritme beregner $\|x\|$.

Algoritme Bits(x)
Inputbetingelse : Heltal $x \geq 0$
Outputkrav : $r = \|x\|$
Metode : $r \leftarrow 0$;
{I} **while** $x > 0$ **do**
  if x ulige **then**
    $x \leftarrow x - 1$;
    $r \leftarrow r + 1$
  $x \leftarrow x/2$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Bits, hvor x_0 angiver den initielle værdi af x.

	Ja	Nej
$r = \|x\|$		
$r + \|x\| = \|x_0\|$		
$\|x\| + 2^r = x_0$		
$r + \|x_0\| = \|x\|$		
$r + \|x\| \leq x_0$		

Opgave 237 (4 %) check

Nedenstående algoritme beregner den heltallige 2-tals-logaritme af n, dvs. $\text{intlog}(n) = \lfloor \log_2 n \rfloor$.

Algoritme Log2(n)
Inputbetingelse : Heltal $n \geq 2$
Outputkrav : $r = \text{intlog}(n) = \lfloor \log_2 n \rfloor$
Metode : $i \leftarrow 1;$
$r \leftarrow 1;$
$p \leftarrow 2;$
$\{I\}$ **while** $2p \leq n$ **do**
  if $p * p \leq n$ **then**
    $p \leftarrow p * p;$
    $r \leftarrow 2 * r$
  else
    $p \leftarrow 2 * p;$
    $r \leftarrow r + 1$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Log2.

	Ja	Nej
$p = 2^r$		
$p = 2r$		
$1 \leq r < p$		
$2p \leq n$		
$p = 2^{\text{intlog}(p)}$		

Opgave 238 (4 %) check

Antag at et array $A[1..n-1]$ indeholder $n-1$ forskellige tal fra mængden $\{1, 2, 3, \ldots, n\}$, hvor $n \geq 2$. Nedenstående algoritme Missing identificerer det element $r \in \{1, 2, 3, \ldots, n\}$ som ikke er i A.

Algoritme Missing($A[1..n-1]$)
Inputbetingelse : Array $A[1..n-1]$ med $n-1 \geq 1$ forskellige heltal fra $\{1, 2, 3, \ldots, n\}$
Outputkrav : $r \in \{1, 2, 3, \ldots, n\} \setminus A$
Metode : $i \leftarrow 1$;
$x \leftarrow 1$;
$y \leftarrow A[1]$;
$\{I\}$ **while** $i < n-1$ **do**
  $i \leftarrow i + 1$;
  $x \leftarrow x + i$;
  $y \leftarrow y + A[i]$
$r \leftarrow n + x - y$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen Missing.

	Ja	Nej
$x = i(i+1)/2$		
$i = i + 1$		
$y = x - i + A[i]$		
$y = \sum_{j=1}^{i} A[j]$		
$1 \leq i \leq n-1$		

Opgave 239 (4 %) check

Antag at et array $A[1..n]$ indeholder $n \geq 1$ heltal. Positionen i i A siges at være *dominerende* hvis $A[i] > A[j]$ for alle $1 \leq j < i$. Lad $\mathrm{dom}(A)$ angive antal dominerende positioner i i A, hvor $1 \leq i \leq n$. Nedenstående algoritme til venstre beregner $\mathrm{dom}(A)$.

Algoritme Domination(A)
Inputbetingelse : Array $A[1..n]$ med $n \geq 1$ heltal
Outputkrav : $r = \mathrm{dom}(A)$
Metode : $i \leftarrow 1$;
$x \leftarrow A[1]$;
$r \leftarrow 1$;
$\{I\}$ **while** $i < n$ **do**
  $i \leftarrow i + 1$;
  if $A[i] > x$ **then**
    $x \leftarrow A[i]$;
    $r \leftarrow r + 1$

For hvert af følgende udsagn, angiv om de er en invariant I for algoritmen.

	Ja	Nej
$r = \mathrm{dom}(A[1..i])$		
$x \geq A[i]$		
$r = \mathrm{dom}(A[1..n])$		
$1 \leq i < n$		
$x = A[i]$		

Opgave 240 (4 %) check

Antag $A[1..n]$ er et sorteret array med n forskellige positive heltal. Lad squares(A) angive antal $A[i]$ hvor $A[i]^2$ også forekommer i A. F.eks. er squares$(1, 3, 4, 7, 9, 16) = 3$, da $1^2 = 1$, $3^2 = 9$ og $4^2 = 16$. Nedenstående algoritme Squares beregner squares(A).

Algoritme Squares($A[1..n]$)
Inputbetingelse : $A[1..n]$ array med n heltal $0 < A[1] < A[2] < \cdots < A[n]$
Outputkrav : $r = \text{squares}(A)$
Metode : $i \leftarrow 1$;
$j \leftarrow 1$;
$r \leftarrow 0$;
$\{I\}$ **while** $j \leq n$ **do**
if $A[i]^2 < A[j]$ **then** $i \leftarrow i + 1$
if $A[i]^2 = A[j]$ **then** $i \leftarrow i + 1$; $j \leftarrow j + 1$; $r \leftarrow r + 1$
if $A[i]^2 > A[j]$ **then** $j \leftarrow j + 1$

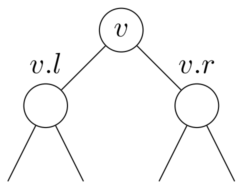
For hvert af følgende udsagn, angiv om de er en invariant I for ovenstående algoritme Squares. Det antages at $A[0] = 0$ og $A[n + 1] = +\infty$.

	Ja	Nej
$i \leq j$		
$r = j - 1$		
$1 \leq i \leq j \leq n$		
$A[i-1]^2 < A[j] \;\wedge\; r = \text{squares}(A[1..j])$		
$A[i-1]^2 < A[j] \;\wedge\; r = \text{squares}(A[1..j-1])$		

Udvidede søgetræer

Opgave 241 (4 %) check

Betragt et rød-sort træ hvor hver knude gemmer et par af heltal ($element$, $vægt$), og parrene er sorteret fra venstre-mod-højre efter stigende $element$ værdi. For en knude v i træet lader vi $v.e$ og $v.w$ betegne parret (e, w) gemt i knuden. Desuden gemmer v værdien $v.W$ som er summen af vægtene i alle knuder i v's undertræ, og $v.prefix$ som er den maksimale sum af vægtene et *præfiks* af parrene i v's undertræ kan have (når parrene sorteres efter element værdi).



Angiv hvorledes $v.prefix$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.prefix = \max(v.l.prefix, v.l.W + v.w, v.l.prefix + v.w + v.r.prefix)$$

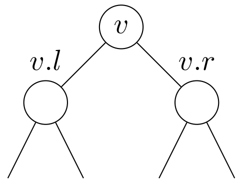
$$v.prefix = \max(v.l.W, v.l.W + v.w, v.l.W + v.w + v.r.W)$$

$$v.prefix = \max(v.l.prefix, v.W + v.r.prefix)$$

$$v.prefix = \max(v.l.prefix, v.l.W + v.w, v.l.W + v.w + v.r.prefix)$$

Opgave 242 (4 %) check

Et interval træ er et rød-sort træ hvor hver knude gemmer præcis ét interval (low, $high$), og intervallerne er sorteret fra venstre-mod-højre efter stigende low værdi. For en knude v i træet ladet vi $v.low$ og $v.high$ betegne endepunkterne på intervallet gemt i knuden. Desuden gemmer v værdien $v.max$ som er den maximale værdi i et interval gemt i v's undertræ.



Angiv hvorledes $v.max$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.max = \max(v.r.max, v.high)$$

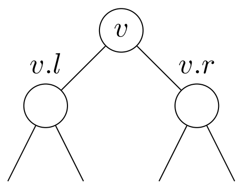
$$v.max = \max(v.l.high, v.high, v.r.high)$$

$$v.max = v.r.max$$

$$v.max = \max(v.r.max, v.l.max, v.high)$$

Opgave 243 (4 %) check

Betragt et søgetræ hvor hver knude v gemmer et tal $v.x$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v to værdier $v.size$ og $v.avg$, som er henholdvis antallet af elementer i v's undertræ og gennemsnittet af alle tallene i v's undertræ.



Angiv hvorledes $v.size$ og $v.avg$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.size = v.l.size + v.r.size$$

$$v.size = v.size + v.l.size + v.r.size$$

$$v.size = 1 + (v.r.size - v.l.size)$$

$$v.size = 1 + v.l.size + v.r.size$$

$$v.avg = (v.x + v.l.avg * v.l.size + v.r.avg * v.r.size)/v.size$$

$$v.avg = (v.x + v.l.avg + v.r.avg)/v.size$$

$$v.avg = (v.x + v.l.avg + v.r.avg)/3$$

$$v.avg = v.x + (v.l.avg + v.r.avg)/2$$

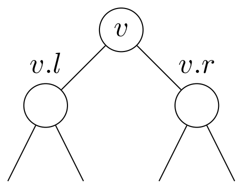
Opgave 244 (4 %) check

Betragt en liste af n punkter $(x_1, y_1), \ldots, (x_n, y_n)$, hvor x_i og y_i er reelle tal og $x_1 < x_2 < \cdots < x_n$. Vi ønsker at finde en sammenhængende række af punkter

$$(x_i, y_i), (x_{i+1}, y_{i+1}), \ldots, (x_{j-1}, y_{j-1}), (x_j, y_j),$$

hvor $i \leq j$, således at $Y_{i,j} = \sum_{k=i}^{j} y_k = y_i + y_{i+1} + \cdots + y_{j-1} + y_j$ er størst mulig. Vi ønsker for en dynamisk liste af punkter at vedligeholde denne maksimale sammenhængende y-sum, *maxysum*.

Betragt et søgetræ hvor hver knude v gemmer et punkt $(v.x, v.y)$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v, der i undertræet indeholer punkterne $(x_k, y_k), \ldots, (x_\ell, y_\ell)$ fire værdier $v.sum = Y_{k,\ell}$, $v.maxysum = \max_{k \leq i \leq j \leq \ell} Y_{i,j}$, $v.pre = \max\{0, \max_{k \leq j \leq \ell} Y_{k,j}\}$, og $v.suf = \max\{0, \max_{k \leq i \leq \ell} Y_{i,\ell}\}$.



Angiv hvorledes $v.sum$ og $v.maxysum$ kan beregnes når den tilsvarende information (incl. *pre* og *suf*) er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.sum = v.l.suf + v.y + v.r.pre$$

$$v.sum = v.l.sum + v.r.sum$$

$$v.sum = v.l.sum + v.y + v.r.sum$$

$$v.sum = v.l.suf + v.r.pre$$

$$v.maxysum = \max\{v.l.maxysum, v.l.suf + v.r.pre, v.r.maxysum\}$$

$$v.maxysum = \max\{v.l.maxysum, v.y, v.r.maxysum\}$$

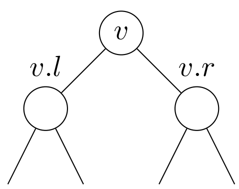
$$v.maxysum = \max\{v.l.maxysum, v.l.suf + v.y + v.r.pre, v.r.maxysum\}$$

$$v.maxysum = \max\{v.l.maxysum, v.r.maxysum\}$$

Opgave 245 (4 %) check

Betragt en liste af n punkter $(x_1, y_1), \ldots, (x_n, y_n)$, hvor x_i og y_i er reelle tal og $x_1 < x_2 < \cdots < x_n$. Et punkt (x_i, y_i) er domineret af et punkt (x_j, y_j) hvis $x_i \leq x_j$ og $y_i \leq y_j$. En liste af punkter er *dominans-fri* hvis og kun hvis intet punkt i listen er domineret af et andet punkt. F.eks. er listen $(3, 7), (5, 3), (7, 6), (13, 4)$ *ikke* dominans-fri, da $(5, 3)$ er domineret af $(13, 4)$. Vi ønsker for en dynamisk liste af punkter at vedligeholde udsagnet om listen er dominans-fri.

Betragt et søgetræ hvor hver knude v gemmer et punkt $(v.x, v.y)$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i v også $v.DF$ og $v.miny$ og $v.maxy$, hvor $v.DF$ er sand hvis og kun hvis mængden af punkterne i v's undertræ er dominans-fri, og $v.miny$ og $v.maxy$ er den mindste og største y værdi gemt i v's undertræ.



Angiv hvorledes $v.miny$ og $v.DF$ kan beregnes når den tilsvarende information (incl. *maxy*) er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.miny = v.l.miny$$

$$v.miny = \min\{v.l.miny, v.y, v.r.miny\}$$

$$v.miny = \min\{v.l.miny, v.r.miny\}$$

$$v.DF = v.l.DF \land v.r.DF \land v.y > v.l.miny \land v.y < v.r.maxy$$

$$v.DF = v.l.DF \land v.r.DF\mathrm{r}$$

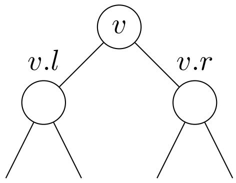
$$v.DF = v.l.DF \land v.r.DF \land v.y < v.l.miny \land v.y > v.r.maxy$$

$$v.DF = v.l.DF \land v.r.DF \land v.y < v.l.miny$$

Opgave 246 (4 %) check

Betragt en liste af n par $(x_1, c_1), \ldots, (x_n, c_n)$, hvor $x_1 < x_2 < \cdots < x_n$ er reelle tal og hvert $c_i \in \{\text{Red}, \text{Green}, \text{Blue}\}$ er en farve.

Betragt et søgetræ hvor hver knude v gemmer et par $(v.x, v.c)$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v, en sandhendsværdi $v.mono$, der angiver om alle elementer i v's undertræ har samme farve, og mængden $v.missing$ af de farver der *ikke* forekommer i v's undertræ.



Angiv hvorledes $v.mono$ og $v.missing$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.mono = v.l.mono \;\wedge\; v.r.mono \;\wedge\; v.l.c = v.r.c \;\wedge\; v.c = v.r.c$$

$$v.mono = v.l.mono \;\wedge\; v.r.mono$$

$$v.mono = v.l.mono \;\wedge\; v.r.mono \;\wedge\; v.l.c = v.r.c$$

$$v.mono = v.l.mono \;\wedge\; v.r.mono \;\wedge\; v.mono$$

$$v.missing = v.l.missing \;\cap\; v.r.missing$$

$$v.missing = \{v.c\} \cup v.l.missing \;\cup\; v.r.missing$$

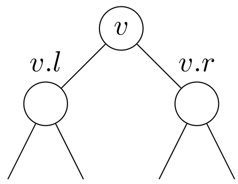
$$v.missing = v.l.missing \;\cup\; v.r.missing$$

$$v.missing = (v.l.missing \;\cap\; v.r.missing) \setminus \{v.c\}$$

Opgave 247 (4 %) check

Givet en streng T indeholdende bogstaver og start- og slut-parenteser `(` og `)`, antages at alle positionerne med parenteser er gemt i et søgetræ, sorteret fra venstre-mod-højre efter stigende position. En knude v gemmer en position $v.p$ og den tilhørende parentes $v.c = T[v.p]$ fra T. For $T =$ ”`a)b(cd(x)dc(a`” gemmes i træet $\langle v.p, v.c\rangle$ parrene $\langle 2, \texttt{)}\rangle$, $\langle 4, \texttt{(}\rangle$, $\langle 7, \texttt{(}\rangle$, $\langle 9, \texttt{)}\rangle$ og $\langle 12, \texttt{(}\rangle$.

Vi ønsker at vedligeholde information om parenteserne er balancerede. I ovenstående eksempel er parenteserne “`) (` $\underline{\texttt{(}}\,\underline{\texttt{)}}$ `(`” ikke balancerede, da kun de markerede parenteser går ud mod hinanden. De restende parenterser “`) ( (`” vil altid være R `)`-parenteser efterfulgt af L `(`-parenteser, hvor $R \geq 0$ og $L \geq 0$. I eksemplet har vi $R = 1$ og $L = 2$. I en knude v i træet gemmes disse værdier $v.R$ og $v.L$ for delsekvensen af parenteserne i v's undertræ.



Angiv hvorledes $v.R$ kan beregnes når $v.c =$ `)` og R og L værdierne er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.R = v.l.R + v.r.R + 1 - v.l.L$$

$$v.R = v.l.R + 1 + v.r.R$$

$$v.R = v.l.R + \max\{0, v.r.R + 1 - v.l.L\}$$

$$v.R = v.l.R + 1 + \max\{0, v.r.R - v.l.L\}$$

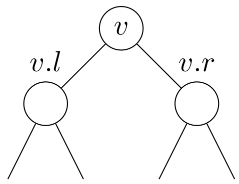
Opgave 248 (4 %) check

For n tal $x_1, \ldots, x_n$ ønsker vi at beregne koefficienterne a og b til polynomiet

$$P(y) = \sum_{i=1}^{n} (x_i - y)^2 = n \cdot y^2 + a \cdot y + b \; .$$

F.eks. for $x_1 = 2$, $x_2 = 3$, og $x_3 = 5$, har vi polynomiet $P(y) = (2-y)^2 + (3-y)^2 + (5-y)^2 = 3y^2 - 20y + 38$, dvs. $a = -20$ og $b = 38$.

Betragt et søgetræ hvor hver knude v gemmer et tal $v.x$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v to værdier $v.a$ og $v.b$, som er a og b koeffienterne for $P(y)$ polynomiet defineret ved alle x værdierne i v's undertræ.



Angiv hvorledes $v.a$ og $v.b$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

$$v.a = v.l.a + v.r.a - 2 * v.x$$

$$v.a = v.l.a * v.r.a * (v.x)^2$$

$$v.a = v.l.a + v.r.a + v.x$$

$$v.a = v.l.a + v.r.a + (v.x)^2$$

$$v.b = v.l.b + v.r.b + 2 * v.x$$

$$v.b = v.l.b + v.r.b + v.x$$

$$v.b = v.l.b + v.r.b + (v.x)^2$$

$$v.b = v.l.b + v.r.b - 2 * v.x$$

Diverse spørgsmål

Opgave 249 (4 %) check

Angiv for hver af følgende algoritmer best-case, worst-case og forventet køretid på input af størrelse n, hvor input kan indeholde identiske elementer.

	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(2^n)$
Forvented tid for MERGE-SORT					
Forvented tid for QUICKSORT					
Best-case tid for MERGE-SORT					
Forvented tid for HEAPSORT					
Worst-case tid for HEAPSORT					
Worst-case tid for QUICKSORT					
Best-case tid for QUICKSORT					
Forvented tid for INSERTION-SORT					
Worst-case tid for MERGE-SORT					
Best-case tid for HEAPSORT					
Worst-case tid for INSERTION-SORT					
Best-case tid for INSERTION-SORT					

Opgave 250 (4 %) check

Angiv worst-case tiden for HEAPSORT på et array med n *identiske* elementer.

$\Theta(n\sqrt{n})$ $\Theta(n^2)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(\sqrt{n})$

Opgave 251 (4 %) check

Angiv hvor mange gange det *største element* i et array med n elementer i worst-case kan blive sammenlignet med andre elementer under udførslen af MERGE-SORT.

$\Theta(n \log n)$ $\Theta(\sqrt{n})$ $\Theta(1)$ $\Theta(n)$ $\Theta(\log n)$

Opgave 252 (4 %) check

Givet et sorteret array $A[1..n]$ ($A[1] < A[2] < \cdots < A[n]$) og et element x, så ønsker vi at finde indekset ℓ således at $A[\ell] \le x < A[\ell+1]$. Det antages at $A[1] \le x < A[n]$. Hvilken af følgende algoritmer er korrekt (kun linierne 2 og 4 varierer i algoritmerne).

$\ell = 1,\ h = n+1$	$\ell = 1,\ h = n+1$	$\ell = 1,\ h = n+1$	$\ell = 1,\ h = n+1$
while $\ell < h$	**while** $\ell < h$	**while** $\ell + 1 < h$	**while** $\ell + 1 < h$
$m = \lfloor (h+\ell)/2 \rfloor$	$m = \lfloor (h+\ell)/2 \rfloor$	$m = \lfloor (h+\ell)/2 \rfloor$	$m = \lfloor (h+\ell)/2 \rfloor$
if $A[m] > x$	**if** $A[m] \le x$	**if** $A[m] \le x$	**if** $A[m] > x$
$\ell = m$	$\ell = m$	$\ell = m$	$\ell = m$
else	**else**	**else**	**else**
$h = m$	$h = m$	$h = m$	$h = m$

Opgave 253 (4 %) check

Strassen's algoritme til multipliaktion af kvadratiske $n \times n$ matricer er en del-og-kombiner algoritme. Angiv hvilken rekursionsligning der beskriver udførselstiden af Strassen's algoritme

$$T(n) \le 7 \cdot T(n/2) + c \cdot n$$

$$T(n) \le 7 \cdot T(n/4) + c \cdot n^2$$

$$T(n) \le 7 \cdot T(n/2) + c \cdot n^2$$

$$T(n) \le 7 \cdot T(n/2) + c \cdot n^3$$

Opgave 254 (4 %) check

Angiv worst-case udførselstiden for hver af følgende algoritmer når input er et array af størrelse n.

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^3)$
INSERTION-SORT							
BINARY-SEARCH							
PARTITION							
QUICKSORT							
HEAPSORT							
MERGE-SORT							

Opgave 255 (4 %) check

Selektionsalgoritmen til at finde det $i$te mindste element i et ikke-sorteret list i worst-case tid $O(n)$ [CLRS, kapitel 9.3], deler input op i grupper af 5 elementer, finder rekursivt medianen af gruppernes medianer, bruger det fundne element som pivot til en opdeling, og kalder rekursivt på en af de to dele. Udførselstiden kan beskrives ved følgende rekursionsligning (afrunding ignoreres):

$$T(n) = T(n/5) + T(7n/10) + c \cdot n$$

Hvad bliver udførselstiden hvis man ændrer grupperne i algoritmen til at have størrelse 3 istedet for 5?

$\Theta(\log n)$ $\Theta(n \log n)$ $\Theta(n)$ $\Theta(n^2)$

Opgave 256 (4 %) check

For en binær max-heap af størrelse n, angiv best-case og worst-case udførselstid for følgende operationer.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(n)$ $\Theta(n \log n)$

Build-Max-Heap, best-case

Insert, worst-case

Heap-Extract-Max, best-case

Heap-Extract-Max, worst-case

Insert, best-case

Build-Max-Heap, worst-case

Opgave 257 (4 %) check

For Quicksort på input af størrelse n, og et givet element e i inputet, hvor mange sammenligninger vil dette element e indgå i under udførslen af QuickSort? Forventet antal sammenligninger er her forventet antal sammenligninger for en tilfældig permutation af input.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$

Forventet antal sammenligninger

Best-case antal sammenligninger

Worst-case antal sammenligninger

Opgave 258 (4 %) check

Hvilke af følgenede udsagn er sande for alle binære søgetræer med n elementer.

	Ja	Nej
Elementet i et blad er altid $\leq$ elementet i roden		
Langs stien fra roden til det største element, er elementerne voksende		
Elementer med samme dybde i træet er sorteret fra venstre-mod-højre		
Det højre barn til en knude med rang r har rang $r+1$		
Længste rod-til-blad sti indeholder $O(\log n)$ knuder		

Opgave 259 (4 %) check

Betragt en $n \times n$ matrix M af heltal, hvor alle rækker og søjler er voksende, dvs. $M[i,j] \leq M[i',j']$ for alle $1 \leq i \leq i' \leq n$ og $1 \leq j \leq j' \leq n$.

Hvad er den bedste worst-case tid man kan opnå for at søge efter et heltal i M?

$O(n)$ $O(\log n)$ $O(\sqrt{n})$ $O(n\sqrt{n})$ $O((\log n)^2)$ $O(n^2)$ $O(n \log n)$

Opgave 260 (4 %) check

Betragt en kø implementeret i et cyklisk array Q, hvor $Q.head$ angiver hovedet af køen og $Q.tail$ er den næste ledige plads i køen (som beskrevet i [CLRS]).

Hvilke af følgende udtryk beregner korrekt størrelsen af køen?

	Ja	Nej
$\lvert Q.tail - Q.head \rvert$		
$\lvert Q.tail - Q.head \rvert \bmod Q.length$		
$Q.tail - Q.head$		
$(Q.tail - Q.head + Q.length) \bmod Q.length$		
$(Q.tail - Q.head + Q.length - 1) \bmod Q.length$		

Opgave 261 (4 %) check

Betragt et vilkårligt binært søgetræ, og lad k være et element i et blad i søgetræet. Lad B være alle elementerne på stien fra roden ned til k, og lad A være alle elementer til venstre for stien, og C alle elementer til højre for stien. Gælder følgende udsagn altid for alle $a \in A$, $b \in B$ og $c \in C$?

	Ja	Nej
$a \leq b$		
$b \leq c$		
$a \leq c$		

Opgave 262 (4 %) check

Givet to søgetræer T_1 og T_2 indeholdende de samme n elementer, så kan T_1 transformeres til T_2 ved en række rotationer. Hvad er worst-case antallet af rotationer der er krævet for at transformere et søgetræ med n elementer til et andet søgetræ T_2 indeholdende de samme n elementer?

$\Theta(\log n)$ $\Theta(n^2)$ $\Theta(n \log n)$ $\Theta(n)$

Opgave 263 (4 %) check

Angiv for hver af nedenstående summer deres værdi i Θ-notationen. Det antages at n er en potens af to.

$\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$

$$\sum_{i=0}^{\log n} 2^i = 1 + 2 + 4 + 8 + \cdots + n$$

$$\sum_{i=1}^{n} \log i = \log 1 + \log 2 + \log 3 + \cdots + \log n$$

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n$$

$$\sum_{i=0}^{\log n} \frac{n}{2^i} = n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} \cdots + \frac{n}{n}$$

$$\sum_{i=1}^{\log n} i\frac{n}{2^i} = 1\frac{n}{2^1} + 2\frac{n}{2^2} + 3\frac{n}{2^3} \cdots + \log n \frac{n}{2^{\log n}}$$

Opgave 264 (4 %) check

Betragt varianten af den binære max-heap, hvor hver knude har d børn istedet for to, og hvor $d \geq 2$ er en parameter. Hvad er tiden for INSERT og EXTRACT-MAX udtrykt som funktion af n og d.

$\Theta(d)$ $\Theta(\log_2 n)$ $\Theta(\log_d n)$ $\Theta(d \log_2 n)$ $\Theta(d \log_d n)$

INSERT

EXTRACT-MAX

Opgave 265 (4 %) check

Betragt en binær max-heap med $n = 2^k - 1$ forskellige elementer, for et heltal $k \geq 1$.

Hvor mange forskellige knuder i max-heapen kan det mindste element være placeret i?

1 n k $k-1$ $2^k - 1$ 2^{k-1}

Opgave 266 (4 %) check

Hvor mange gange kaldes RANDOMIZED-PARTITION i RANDOMIZED-SELECT på et array af størrelse n?

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$
Flest antal kald (worst case)					
Forventede antal kald					
Færrest antal kald (best case)					

Opgave 267 (4 %) check

Lad v være en knude i et rød-sort søgetræ, og antag der er n elementer i v's venstre undertræ. Hvor mange elementer kan der så maksimalt være i v's højre undertræ, dvs. hvor ubalanceret kan en knude v være i et rød-sort søgetræ?

$\Theta(n^2)$ $\Theta(2^n)$ $\Theta(n\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$

Opgave 268 (4 %) check

Antag et array $A[1..n]$ repræsenterer en binær max-heap indeholdende n elementer. Hvor hurtigt kan man konstruere et søgetræ (ikke nødvendigvis balanceret) indeholdende elementerne $A[1..n]$?

$\Theta(n^2)$ $\Theta(\log n)$ $\Theta(n)$ $\Theta(n \log n)$

Løkke opgaver

Opgave 269

check

Algoritme loop1(n)
  $i = 1$
  while $i \leq n$
    $j = 1$
    while $j \leq i$
      $j = 2 * j$
    $i = 2 * i$

Algoritme loop2(n)
  $i = 1$
  while $i \leq n$
    $j = 1$
    while $j \leq n$
      $j = 2 * j$
    $i = 2 * i$

Algoritme loop3(n)
  $i = 1$
  while $i \leq n$
    $j = i$
    while $j \leq n$
      $j = 2 * j$
    $i = 2 * i$

Algoritme loop4(n)
  $i = n$
  while $i > 0$
    $j = i$
    while $j > 0$
      $j = \lfloor j/2 \rfloor$
    $i = \lfloor i/2 \rfloor$

Algoritme loop5(n)
  $s = 1$
  for $i = 1$ **to** n
    $j = 1$
    while $j \leq s$
      $j = j + 1$
    $s = 2 * s$

Algoritme loop6(n)
  $s = 1$
  for $i = 1$ **to** n
    $j = s$
    while $j > 0$
      $s = s + 1$
      $j = j - 1$

Algoritme loop7(n)
  $i = 1$
  $p = 1$
  while $p \leq n$
    $i = i + 1$
    $p = p * i$

Algoritme loop8(n)
  $i = 1$
  $j = n$
  while $i \leq j$
    $i = 4 * i$
    $j = 2 * j$

Algoritme loop9(n)
  $i = 1$
  $j = n$
  while $i \leq j$
    $i = i * 2$
    $j = \lfloor j/2 \rfloor$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n^2)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta((\log n)^2)$	$\Theta(n \log n)$	$\Theta(\frac{\log n}{\log \log n})$	$\Theta(2^n)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 270

check

```
Algoritme loop1(n)          Algoritme loop2(n)          Algoritme loop3(n)
  i = 1                       i = 1                       i = 1
  while i * i ≤ n             while i ≤ n                 while i ≤ n
      i = i + i                   i = 2 * i                   i = 3 * i

Algoritme loop4(n)          Algoritme loop5(n)          Algoritme loop6(n)
  i = 1                       i = 1                       i = 1
  while i ≤ n                 while i ≤ n * n             while i ≤ n * n
      i = i + i                   i = 2 * i                   i = 3 * i

Algoritme loop7(n)          Algoritme loop8(n)          Algoritme loop9(n)
  i = n                       s = n                       i = 2
  while i > 0                 while s > 0                 while i ≤ n
      if i ulige then             s = ⌊s/2⌋                   i = i * i
          i = i − 1
      else
          i = i/2
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(\log\log n)$	$\Theta(\log n)$	$\Theta((\log n)^2)$	$\Theta(n^2)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n\log n)$	$\Theta(n)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 271 check

Algoritme loop1(n)
```
s = 2
while s ≤ n
    s = s * s
```

Algoritme loop2(n)
```
i = 0
s = 0
q = 0
while q ≤ n
    i = i + 1
    s = s + i
    q = q + s
```

Algoritme loop3(n)
```
i = 0
s = 0
while s ≤ n
    i = i + 1
    s = s + i
```

Algoritme loop4(n)
```
i = 1
j = 1
s = 0
while s ≤ n
    while j ≤ s
        j = 2 * j
    s = s + i
    i = i + 1
```

Algoritme loop5(n)
```
j = n
i = 1
while j ≥ 0
    j = j - i
    i = i + 1
```

Algoritme loop6(n)
```
s = 0
i = 1
while s ≤ n
    s = s + i
    i = i + 1
```

Algoritme loop7(n)
```
i = 1
while i ≤ n
    j = 1
    k = 1
    while k ≤ n
        j = j + 1
        k = k + j
    i = 2 * i
```

Algoritme loop8(n)
```
for i = 1 to n
    j = i
    while j ≤ n
        j = 2 * j
```

Algoritme loop9(n)
```
i = 0
j = n
while i ≤ j
    i = i + 1
    j = j - 1
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(\sqrt{n}\log n)$	$\Theta(\log\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(\log n)$	$\Theta((\log n)^2)$	$\Theta(\sqrt[3]{n})$	$\Theta(n\log n)$	$\Theta(n^2)$
loop1									
loop2									
loop3									
loop4									
loop5									
loop6									
loop7									
loop8									
loop9									

Opgave 272 check

Algoritme loop1(n)
  $i = 1$
  $j = 0$
  while $i \leq n$
    $i = i + i$
    while $j < i$
      $j = j + 1$

Algoritme loop2(n)
  $i = 1$
  $j = 1$
  while $i \leq n$
    while $j \leq i$
      $j = j + 1$
    $i = 2 * i$

Algoritme loop3(n)
  $i = 1$
  $s = 0$
  while $s \leq n$
    $j = 1$
    while $j \leq i$
      $j = j + 1$
    $s = s + i$
    $i = i + 1$

Algoritme loop4(n)
  $i = 1$
  $s = 1$
  while $s \leq n * n$
    $i = i + 1$
    $s = s + i$

Algoritme loop5(n)
  $i = 1$
  while $i \leq n$
    $j = 0$
    while $j \leq i$
      $j = j + 1$
    $i = 2 * i$

Algoritme loop6(n)
  $i = 1$
  while $i \leq n$
    $j = 0$
    while $j \leq n$
      $j = j + i$
    $i = 2 * i$

Algoritme loop7(n)
  $i = 1$
  while $i \leq n$
    $j = 1$
    while $j \leq i$
      $j = j + 1$
    $i = 2 * i$

Algoritme loop8(n)
  $i = n$
  while $i > 0$
    $i = i - 1$

Algoritme loop9(n)
  $i = n$
  while $i \geq 1$
    $j = i$
    while $j \leq n$
      $j = 2 * j$
    $i = i - 1$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n \log n)$	$\Theta(\log n)$	$\Theta((\log n)^2)$	$\Theta(2^n)$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n} \log n)$	$\Theta(n^2)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 273 check

Algoritme loop1(n)
  $s = 0$
  $i = 1$
  while $i * i \leq n$
    for $j = 1$ **to** i
      $s = s + 1$
    $i = i + 1$

Algoritme loop2(n)
  $s = 0$
  $i = n$
  while $i > 1$
    for $j = 1$ **to** i
      $s = s + 1$
    $i = \lfloor i/2 \rfloor$

Algoritme loop3(n)
  $s = 1$
  for $i = 1$ **to** n
    $s = s + 1$

Algoritme loop4(n)
  $s = 1$
  for $i = n$ **to** 1 **step** -1
    $s = s + 1$

Algoritme loop5(n)
  $s = 1$
  $i = 1$
  while $i \leq n$
    for $j = 1$ **to** i
      $s = s + 1$
    $i = 2 * i$

Algoritme loop6(n)
  $s = 1$
  while $s \leq n$
    $s = s + 1$

Algoritme loop7(n)
  for $i = 1$ **to** n
    $j = 0$
    while $j \leq n$
      $j = j + i$

Algoritme loop8(n)
  for $i = 1$ **to** n
    $j = 1$
    while $j \leq i$
      $j = 2 * j$

Algoritme loop9(n)
  for $i = 1$ **to** n
    $j = 1$
    while $j \leq n$
      $j = 2 * j$

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\sqrt[3]{n})$	$\Theta(n \log n)$	$\Theta(\sqrt{n})$	$\Theta(\log \log n)$	$\Theta(n^2)$	$\Theta((\log n)^2)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 274

check

```
Algoritme loop1(n)
  for i = 1 to n
    j = i
    while j > 1
      j = ⌊j/2⌋
```

```
Algoritme loop2(n)
  i = 0
  while i ≤ n
    j = i
    while j > 0
      j = ⌊j/2⌋
    i = i + 1
```

```
Algoritme loop3(n)
  i = 1
  j = 1
  s = 0
  while i ≤ n
    if i = j then
      for k = 1 to n
        s = s + 1
      j = 2 * j
    i = i + 1
```

```
Algoritme loop4(n)
  i = 1
  s = 0
  while i ≤ n
    for j = i to n
      s = s + 1
    i = i + i
```

```
Algoritme loop5(n)
  i = 1
  while i ≤ n
    j = 0
    while j ≤ n
      j = j + 1
    i = 2 * i
```

```
Algoritme loop6(n)
  i = 1
  while i ≤ n
    j = 1
    while j ≤ i
      j = 2 * j
    i = i + 1
```

```
Algoritme loop7(n)
  i = 1
  while i ≤ n
    j = i
    while j ≤ n
      j = j + 1
    i = 2 * i
```

```
Algoritme loop8(n)
  i = 1
  while i ≤ n
    j = n
    while j > 1
      j = j - 1
    i = 2 * i
```

```
Algoritme loop9(n)
  s = 0
  i = n
  while i > 1
    for j = 1 to n
      s = s + 1
    i = ⌊i/2⌋
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n \log n)$	$\Theta(n)$	$\Theta(n^2)$	$\Theta(\sqrt{n})$	$\Theta((\log n)^2)$	$\Theta(\log n)$	$\Theta(\sqrt{n} \log n)$	$\Theta(n\sqrt{n})$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 275 check

```
Algoritme loop1(n)
  for i = 0 to n
    j = 0
    s = 0
    while s ≤ i
      j = j + 1
      s = s + j
```

```
Algoritme loop2(n)
  for i = 1 to n
    j = 1
    while j ≤ i
      j = j + 1
```

```
Algoritme loop3(n)
  for i = 1 to n
    j = i
    while j > 0
      j = j − 1
```

```
Algoritme loop4(n)
  i = 0
  j = 0
  while i ≤ n
    if i < j then
      i = i + 1
    else
      j = j + 1
      i = 0
```

```
Algoritme loop5(n)
  i = 1
  while i ≤ n
    j = 1
    while j ≤ i
      j = j + 1
    i = i + 1
```

```
Algoritme loop6(n)
  i = 1
  while i ≤ n
    j = 1
    while j ≤ n
      j = j + 1
    i = i + 1
```

```
Algoritme loop7(n)
  s = 0
  for i = 1 to n
    for j = 1 to n
      if i = j then
        for k = 1 to n
          s = s + 1
```

```
Algoritme loop8(n)
  s = 0
  i = n
  while i > 0
    for j = 1 to i
      s = s + 1
    i = i − 1
```

```
Algoritme loop9(n)
  s = 1
  for i = 1 to n
    for j = 1 to n
      s = s + 1
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n\sqrt{n})$	$\Theta(\log n)$	$\Theta(n \log n)$	$\Theta(\sqrt{n})$	$\Theta(n^2 \cdot \log n)$	$\Theta((\log n)^2)$	$\Theta(n)$	$\Theta(n^2)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								
loop9								

Opgave 276 check

```
Algoritme loop1(n)
  s = 1
  for i = 1 to n
      for j = 1 to n
          s = s + 1
      for k = 1 to n
          s = s + 1
```

```
Algoritme loop2(n)
  s = 1
  for i = 1 to n
      for j = i to n
          s = s + 1
```

```
Algoritme loop3(n)
  s = 1
  for i = n to 1 step −1
      for j = n to 1 step −1
          s = s + 1
```

```
Algoritme loop4(n)
  for i = 1 to n
      for j = 1 to i
          k = 1
          while k ≤ i + j
              k = 2 * k
```

```
Algoritme loop5(n)
  s = 0
  for i = 1 to n
      for j = 1 to i * i
          s = s + 1
```

```
Algoritme loop6(n)
  s = 0
  for i = 1 to n
      for j = 1 to n
          for k = 1 to n
              s = s + 1
```

```
Algoritme loop7(n)
  s = 0
  for i = 1 to n
      for j = i to n
          for k = i to j
              s = s + 1
```

```
Algoritme loop8(n)
  s = 0
  j = 0
  for i = 1 to n
      j = j + i
      for k = 1 to j
          s = s + 1
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta((\log n)^2)$	$\Theta(\sqrt{n})$	$\Theta(n^3)$	$\Theta(n)$	$\Theta(n^2 \cdot \log n)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta(n \log n)$
loop1								
loop2								
loop3								
loop4								
loop5								
loop6								
loop7								
loop8								