

Indhold

Asymptotisk notation	2
Analyse af løkker	7
Indsættelse i søgetræer	13
Max-Heap-Insert	18
Build-Max-Heap	22
Heap-Extract-Max	27
Partition	32
Radix-sort	36
Lineær probing	39
Kvadratisk probing	44
Dobbelt hashing	49
Lovlige rød-sortede træer	54
Rød-sort træ indsættelse	59
Union-find	65
Huffman koder	72
Rekursionsligninger	77
BFS	79
Lovlige BFS træer	84
DFS	89
Dijkstras algoritme	98
Prims algoritme	103
Kruskals algoritme	108
Topologisk sortering	113
Stærke sammenhængskomponenter	118
Løkke opgaver	122
Amortiseret analyse	131
Invarianter	137
Udvidede søgetræer	147
Diverse spørgsmål	155

Asymptotisk notation

Opgave 1 (6 %)

check

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

	Ja	Nej
$(\log n)^3 = O(n^2)$		1.1
$4 \log(n^2) = O(n^{3/2})$		1.2
$n^2 + 3 \log n = O(\log n)$		1.3
$n^2 = O(n)$		1.4
$2n \cdot \log n = O(2^n)$		1.5
$3^3 + \sqrt{n} = O(n!)$		1.6
$3\sqrt{n} + 7n^{1/3} = O(n^{0.1})$		1.7
$4(\log n)^4 = O(2^n)$		1.8
$n \cdot \log n = O(\sqrt{n})$		1.9
$\log n = \Omega(\sqrt{n})$		1.10
$n \cdot \log n = \Omega(n^{0.1})$		1.11
$n^{0.01}/5 = \Omega(\log n)$		1.12

Opgave 2 (6 %)

check

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

	Ja	Nej
$6\sqrt{n} \cdot \log n = O(n^{2/3})$		2.1
$n^{0.1} = O(n)$		2.2
$7(\log n)^2 = O(n^{0.001})$		2.3
$n \cdot \log n = O(\sqrt{n} \cdot \log n)$		2.4
$2^n = O(n^3)$		2.5
$n^2 \log n = O(\log n)$		2.6
$n^3 = O(\sqrt{n})$		2.7
$n = O(\log(n!))$		2.8
$7n \cdot \log n = O(n^{0.001})$		2.9
$n \cdot \log n = \Omega(8^{\log n})$		2.10
$\sqrt{n} = \Omega((\log n)^2)$		2.11
$n \cdot \log n = \Omega(n^{2/3})$		2.12

Opgave 3 (6 %)

check

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

Ja Nej

$$\sqrt{n} = O(n^{2/3}) \quad 3.1$$

$$\log(n^2) = O(n \cdot \log n) \quad 3.2$$

$$n = O(n^{0.01}) \quad 3.3$$

$$4n = O(n) \quad 3.4$$

$$8^{\log n} = O(\log n) \quad 3.5$$

$$2^n = O(\log(n^2)) \quad 3.6$$

$$\log(n!) = O(n) \quad 3.7$$

$$n^2 = O(n^{0.001}) \quad 3.8$$

$$(\log n)^2 = O(n^{0.01}) \quad 3.9$$

$$n^{0.001} = \Omega(\log(n^2)) \quad 3.10$$

$$n^2 \log n = \Omega(\sqrt{n}) \quad 3.11$$

$$3^n = \Omega(\log n) \quad 3.12$$

Opgave 4 (6 %)

check

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

Ja Nej

$$n + \sqrt{n} = O(n \cdot \log n) \quad 4.1$$

$$2^{3 \log n} = O(n^n) \quad 4.2$$

$$7n^3 = O(3^3) \quad 4.3$$

$$n^2 \log n = O(\sqrt{n}) \quad 4.4$$

$$n \cdot \log n = O(n^3) \quad 4.5$$

$$\log(n!) = O(n^3) \quad 4.6$$

$$2^n = O(2^{3 \log n}) \quad 4.7$$

$$5n^{2/3} \cdot n^{1/3} = O(n^n) \quad 4.8$$

$$\sqrt{n} = O(2^{\log n}) \quad 4.9$$

$$\log n + \log n = \Theta(\log n) \quad 4.10$$

$$4\sqrt{n} \cdot \log n = \Omega(\sum_{i=1}^n i) \quad 4.11$$

$$(\log n)^3 = \Theta((\log n)^6) \quad 4.12$$

Opgave 5 (6 %)

check

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

Ja Nej

$\log(n!) = O(n!)$	5.1
$2^n = O(n^{0.1})$	5.2
$n^{3/2} = O(n^2 \log n)$	5.3
$6n^2 = O(2^{\log n})$	5.4
$\log n = O(\sqrt{n} \cdot \log n)$	5.5
$5 \log n + n^2 = O(\log(n^2))$	5.6
$2^{3 \log n} = O(n)$	5.7
$1 = O(n^{0.1})$	5.8
$n \cdot \log n = O(n)$	5.9
$1 = \Omega(n^{1/3})$	5.10
$6 \log n = \Theta(4^{\log n})$	5.11
$n^{0.001} = \Omega(\sqrt{n} \cdot \log n)$	5.12

Opgave 6 (6 %)

check

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

Ja Nej

$\log n = O(2^{\log n})$	6.1
$8^{\log n} = O(n^{2/3})$	6.2
$\sqrt{n} = O(n^{2/3})$	6.3
$2^{\log n} = O(2^{2 \log n})$	6.4
$\log(n^2) = O(\log n)$	6.5
$\sqrt{n} \cdot \log n = O(\sqrt{n})$	6.6
$\log(n!)/7 = O(n^2 \log n)$	6.7
$n! + \sqrt{n} \cdot (\log n)/3 = O(\sqrt{n})$	6.8
$\sqrt{n} = O((\log n)^3)$	6.9
$5\sqrt{n} + 2 \cdot 8^{\log n} = \Omega(n!)$	6.10
$8^{\log n} = \Theta(2^{3 \log n})$	6.11
$n \cdot \log n = \Omega(2^{3 \log n})$	6.12

Opgave 7 (6 %)

check

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

Ja Nej

$1 + \log(n^2)/2 = O(\log n)$	7.1
$2n = O(n \cdot \log n)$	7.2
$1 + 3^3 = O(2^{3 \log n})$	7.3
$7n \cdot \log n = O(\log n)$	7.4
$n! = O((\log n)^3)$	7.5
$n^2 \log n = O(\sqrt{n})$	7.6
$n \cdot \log n = O(\log(n^2))$	7.7
$n^{2/3} = O(2^{3 \log n})$	7.8
$3n \cdot \log n = O(1)$	7.9
$n = \Theta(n^{0.1})$	7.10
$\log n = \Omega(n \cdot \log n)$	7.11
$4 \log n = \Theta(\log n)$	7.12

Opgave 8 (6 %)

check

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

Ja Nej

$n \cdot \log n = O(\sqrt{n})$	8.1
$n^{0.01} = O(\sqrt{n} \cdot \log n)$	8.2
$4 \cdot 3^n = O(\log n)$	8.3
$n^{2/3} = O(n^{2/3} \cdot n^{1/3})$	8.4
$5^5 + n^{0.001} = O(n \cdot \log n)$	8.5
$\log(n^2) = O(\log n)$	8.6
$5n^{1/3} = O(\log n)$	8.7
$2^{3 \log n} = O(n^3)$	8.8
$6\sqrt{n} \cdot \log n = O(\sqrt{n})$	8.9
$n/6 = \Theta(n)$	8.10
$2^{2 \log n} = \Theta(n\sqrt{n})$	8.11
$n^3 = \Theta(n \cdot \log n)$	8.12

Opgave 9 (6 %)

check

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

Ja Nej

$\sqrt{n}/2 = O(\log(n^2))$	9.1
$2^n = O(n \cdot \log n)$	9.2
$6 \log(n!) + n \cdot \log n = O(2^{\log n})$	9.3
$\log n = O(\sqrt{n} \cdot \log n)$	9.4
$2 = O((\log n)^2)$	9.5
$(\log n)^3 = O(n \cdot \log n)$	9.6
$n \cdot \log n = O(2^n)$	9.7
$n^3 + n \cdot \log n = O(n \cdot \log n)$	9.8
$8^{\log n} = O(2^n)$	9.9
$n^{0.1}/6 = \Theta(n^n)$	9.10
$4^{\log n} = \Theta(\sum_{i=1}^n i)$	9.11
$1 = \Theta(5^5)$	9.12

Opgave 10 (6 %)

check

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

Ja Nej

$n \cdot \log n = O(2^{2 \log n})$	10.1
$2^{3 \log n} = O(n!)$	10.2
$n = O(n \cdot \log n)$	10.3
$n = O(n\sqrt{n})$	10.4
$3n = O(n)$	10.5
$\sqrt{n} + \sqrt{n} = O(n^{0.1})$	10.6
$\log(n!) + 6n^{3/2} = O(\log n)$	10.7
$n^{0.1} = O(\sqrt{n} \cdot \log n)$	10.8
$n \cdot (\log n)/3 = O(2^n)$	10.9
$n \cdot \log n = \Theta(\log(n!))$	10.10
$1 + 4 = \Theta(4^4)$	10.11
$\sqrt{n} \cdot \log n = \Omega(\sqrt{n})$	10.12

Opgave 13 (6 %)

check

Algorithm loop1(n) $s = 0$ for $i = 1$ to n for $j = 1$ to $i * i$ $s = s + 1$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $i = 2 * i$
Algorithm loop3(n) $i = 1$ while $i \leq n * n$ $i = 2 * i$	Algorithm loop4(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq i$ $j = 2 * j$ $i = 2 * i$

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

$\Theta(n^2)$	$\Theta((\log n)^2)$	$\Theta(\log n)$	$\Theta(n^3)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(\sqrt{n})$	$\Theta(n^2 \cdot \log n)$
loop1							13.1
loop2							13.2
loop3							13.3
loop4							13.4

Opgave 14 (6 %)

check

Algorithm loop1(n) $i = 1$ while $i \leq n$ $i = 2 * i$	Algorithm loop2(n) $i = n$ while $i > 0$ $i = i - 1$
Algorithm loop3(n) $i = 1$ while $i \leq n * n$ $i = 3 * i$	Algorithm loop4(n) $i = 1$ $j = 1$ $s = 0$ while $s \leq n$ while $j \leq s$ $j = 2 * j$ $s = s + i$ $i = i + 1$

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

$\Theta(n^2)$	$\Theta(2^n)$	$\Theta(\sqrt{n})$	$\Theta(n^3)$	$\Theta(\log n)$	$\Theta(n \log n)$	$\Theta(n)$	$\Theta((\log n)^2)$
loop1							14.1
loop2							14.2
loop3							14.3
loop4							14.4

Opgave 15 (6 %)

check

Algorithm loop1(n) $s = 0$ for $i = 1$ to n for $j = 1$ to n for $k = 1$ to n $s = s + 1$	Algorithm loop2(n) for $i = 1$ to n $j = i$ while $j > 0$ $j = j - 1$
Algorithm loop3(n) $i = 1$ while $i \leq n$ $j = 0$ while $j \leq n$ $j = j + 1$ $i = 2 * i$	Algorithm loop4(n) $i = n$ $j = 0$ while $i > 0$ if $j < i$ $j = j + 1$ else $j = 0$ $i = i - 1$

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

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

loop1	15.1
loop2	15.2
loop3	15.3
loop4	15.4

Opgave 17 (6 %)

check

Algorithm loop1(n) $s = 1$ for $i = 1$ to $n * n$ for $j = 1$ to n $s = s + 1$	Algorithm loop2(n) $s = 1$ for $i = 1$ to n $s = s + 1$
Algorithm loop3(n) $s = 0$ $i = n$ while $i > 0$ for $j = 1$ to i $s = s + 1$ $i = i - 1$	Algorithm loop4(n) $i = n$ $j = 0$ while $i > 0$ if $j < i$ $j = j + 1$ else $j = 0$ $i = i - 1$

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

$\Theta(\sqrt{n} \log n)$	$\Theta((\log n)^2)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$
loop1							17.1
loop2							17.2
loop3							17.3
loop4							17.4

Opgave 18 (6 %)

check

Algorithm loop1(n) $s = 0$ for $i = 1$ to n for $j = 1$ to n for $k = 1$ to n $s = s + 1$	Algorithm loop2(n) $i = n$ while $i > 0$ $i = i - 1$
Algorithm loop3(n) $i = 2$ while $i \leq n$ $i = i * i$	Algorithm loop4(n) for $i = 1$ to n $j = i$ while $j \leq n$ $j = 2 * j$

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

$\Theta(n^2)$	$\Theta(n)$	$\Theta(\log \log n)$	$\Theta(\sqrt{n})$	$\Theta((\log n)^2)$	$\Theta(n \log n)$	$\Theta(n^3)$	$\Theta(\log n)$
loop1							18.1
loop2							18.2
loop3							18.3
loop4							18.4

Opgave 19 (6 %)

check

Algorithm loop1(n) for $i = 1$ to n $j = i$ while $j > 0$ $j = j - 1$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $i = 2 * i$
Algorithm loop3(n) $s = 0$ $i = 1$ while $s \leq n$ $s = s + i$ $i = i + 1$	Algorithm loop4(n) $i = 1$ while $i \leq n$ $j = i$ while $j \leq n$ $j = j + 1$ $i = 2 * i$

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

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

loop1	19.1
loop2	19.2
loop3	19.3
loop4	19.4

Opgave 20 (6 %)

check

Algorithm loop1(n) for $i = 1$ to n $j = i$ while $j > 0$ $j = j - 1$	Algorithm loop2(n) $s = 1$ for $i = n$ to 1 step -1 for $j = n$ to 1 step -1 $s = s + 1$
Algorithm loop3(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq i$ $j = 2 * j$ $i = i + 1$	Algorithm loop4(n) $i = 1$ $j = n$ while $i \leq j$ $i = 4 * i$ $j = 2 * j$

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

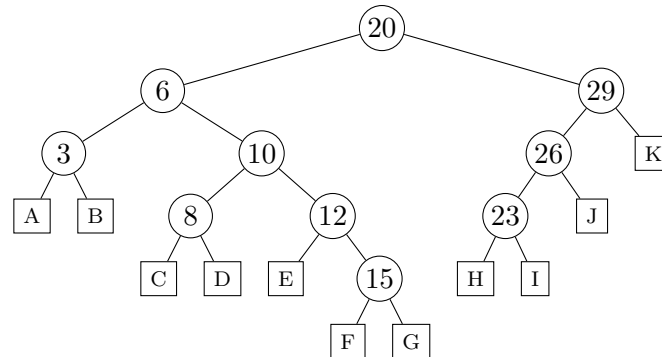
$\Theta(n)$ $\Theta(n^3)$ $\Theta(\log n)$ $\Theta(\frac{\log n}{\log \log n})$ $\Theta(2^n)$ $\Theta((\log n)^2)$ $\Theta(n \log n)$ $\Theta(n^2)$

loop1	20.1
loop2	20.2
loop3	20.3
loop4	20.4

Indsættelse i søgetræer

Opgave 21 (4 %)

check

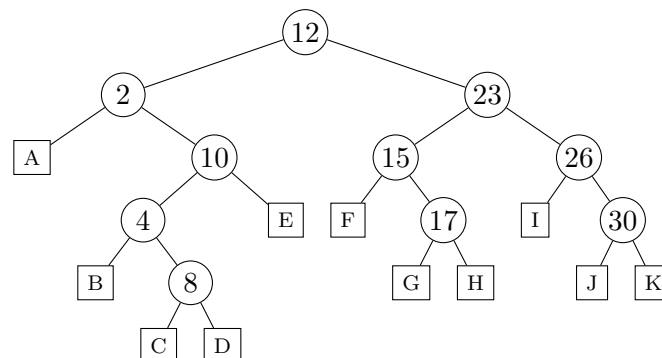


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 18, 2, 11, 25 and 22 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 22 (4 %)

check

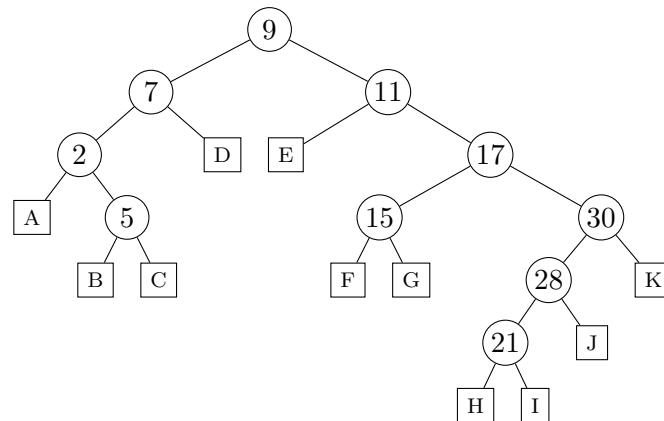


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 19, 28, 27, 1 and 21 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 23 (4 %)

check

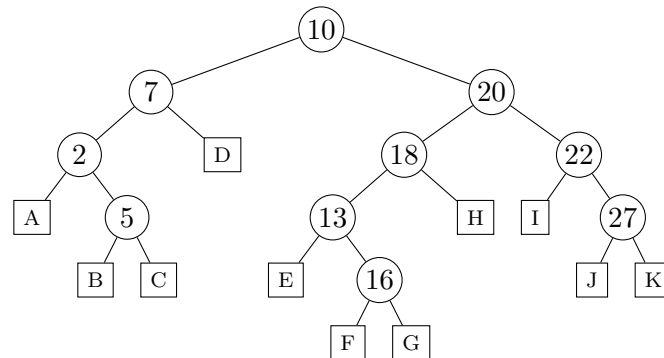


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 10, 3, 22, 25 and 20 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 24 (4 %)

check

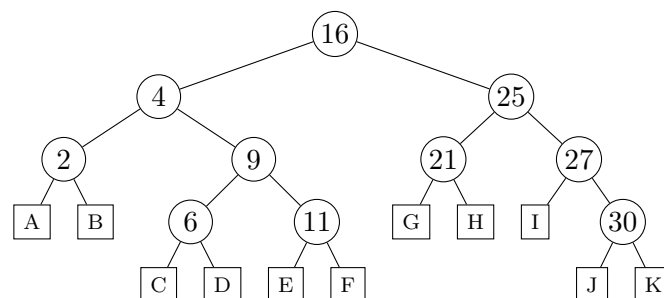


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 11, 19, 17, 28 og 1 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 25 (4 %)

check

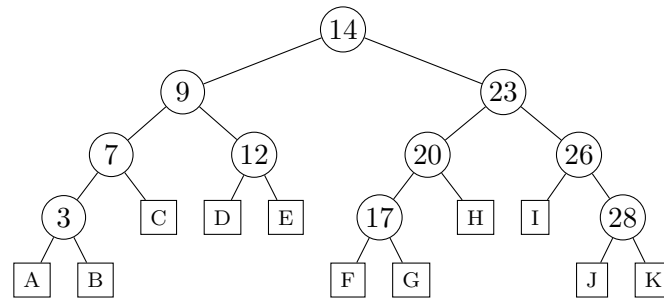


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 18, 5, 12, 19 and 22 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 26 (4 %)

check



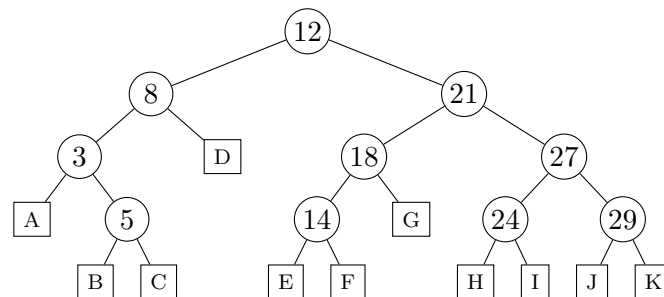
Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 16, 2, 4, 19 and 22 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(16)	26.1
INSERT(2)	26.2
INSERT(4)	26.3
INSERT(19)	26.4
INSERT(22)	26.5

Opgave 27 (4 %)

check



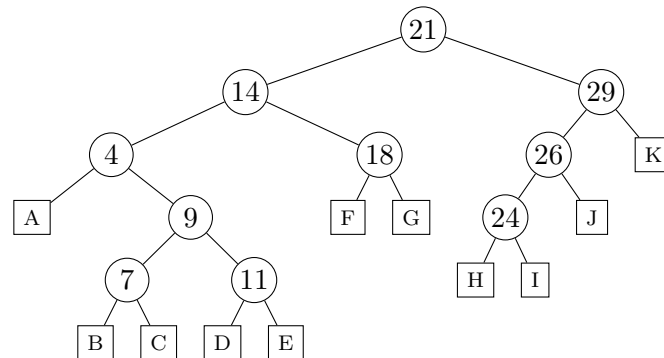
Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 17, 10, 19, 15 and 23 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(17)	27.1
INSERT(10)	27.2
INSERT(19)	27.3
INSERT(15)	27.4
INSERT(23)	27.5

Opgave 28 (4 %)

check

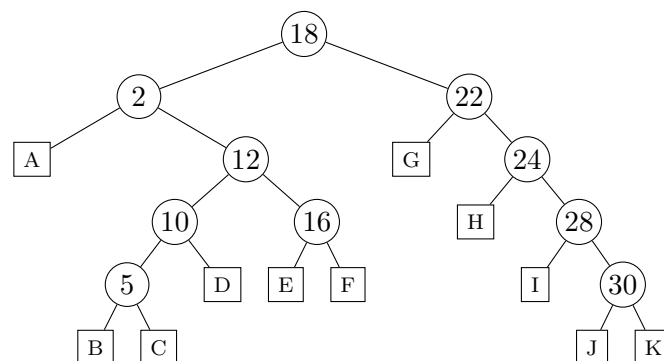


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 15, 20, 28, 16 and 12 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 29 (4 %)

check

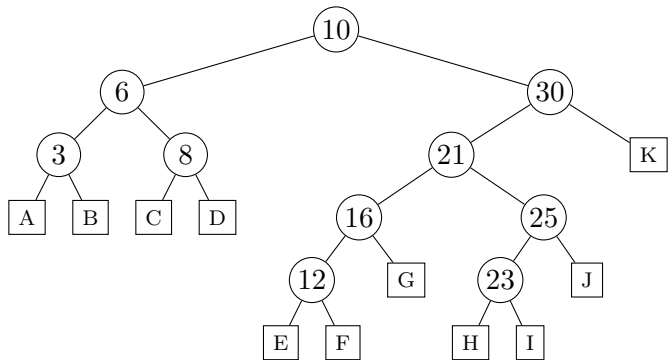


Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 26, 29, 21, 8 og 6 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

[illegible]

Opgave 30 (4 %)

check



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 18, 7, 4, 24 and 9 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(18)												30.1
INSERT(7)												30.2
INSERT(4)												30.3
INSERT(24)												30.4
INSERT(9)												30.5

Max-Heap-Insert

Opgave 31 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 3, 11, 13, 14, 1, 8 og 5 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
14	13	11	3	1	8	5

1	2	3	4	5	6	7
14	11	13	3	1	8	5

1	2	3	4	5	6	7
3	11	13	14	1	8	5

1	2	3	4	5	6	7
14	13	11	8	5	3	1

1	2	3	4	5	6	7
13	14	8	11	1	3	5

Opgave 32 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 2, 7, 4, 9, 8, 11 og 3 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

Opgave 33 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 10, 2, 1, 13, 12, 14 og 6 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
14	13	10	2	12	1	6
1	2	3	4	5	6	7
14	12	13	2	10	1	6
1	2	3	4	5	6	7
14	13	12	10	6	2	1
1	2	3	4	5	6	7
10	13	14	2	12	1	6
1	2	3	4	5	6	7
10	2	1	13	12	14	6

Opgave 34 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 10, 12, 1, 14, 6, 11 og 13 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
10	12	1	14	6	11	13
1	2	3	4	5	6	7
14	13	12	11	10	6	1
1	2	3	4	5	6	7
12	14	13	10	6	11	1
1	2	3	4	5	6	7
14	12	13	10	6	1	11
1	2	3	4	5	6	7
14	12	13	10	6	11	1

Opgave 35 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 4, 10, 1, 2, 5, 12 og 3 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
12	10	5	4	3	2	1

1	2	3	4	5	6	7
12	10	4	2	5	1	3

1	2	3	4	5	6	7
12	5	10	2	4	1	3

1	2	3	4	5	6	7
10	5	12	2	4	1	3

1	2	3	4	5	6	7
4	10	1	2	5	12	3

Opgave 36 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 6, 3, 9, 11, 12, 8 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
6	3	9	11	12	8	14

1	2	3	4	5	6	7
9	12	14	11	3	8	6

1	2	3	4	5	6	7
14	12	11	9	8	6	3

1	2	3	4	5	6	7
14	11	12	3	9	6	8

1	2	3	4	5	6	7
14	12	9	11	3	8	6

Opgave 37 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 3, 12, 4, 10, 11, 2 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
14	12	4	10	11	2	3

1	2	3	4	5	6	7
3	12	4	10	11	2	14

1	2	3	4	5	6	7
14	11	12	3	10	2	4

1	2	3	4	5	6	7
12	11	14	10	3	2	4

1	2	3	4	5	6	7
14	12	11	10	4	3	2

Opgave 38 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 6, 8, 10, 14, 7, 1 og 12 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
14	8	12	6	7	1	10
1	2	3	4	5	6	7
14	12	10	8	7	6	1
1	2	3	4	5	6	7
10	14	12	8	7	1	6
1	2	3	4	5	6	7
14	10	12	6	7	1	8
1	2	3	4	5	6	7
6	8	10	14	7	1	12

Opgave 39 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 2, 8, 1, 5, 10, 9 og 6 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

Opgave 40 (4 %)

check

Angiv den binære max-heap efter indsættelse af elementerne 10, 5, 6, 9, 8, 11 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

Build-Max-Heap

Opgave 41 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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
6	7	9	8	5	2	4	1	3

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	2	5	4	1	3

Opgave 42 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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
4	8	9	6	2	7	1	5	3

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

Opgave 43 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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

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

Opgave 44 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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

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

Opgave 45 (4 %)

check

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

Hvad er resultatet af at anvende 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
7	8	9	6	1	5	2	3	4

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

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

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

Opgave 46 (4 %)

check

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

Hvad er resultatet af at anvende 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
8	6	9	4	5	2	7	3	1

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	6	8	4	5	2	7	1	3

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

Opgave 47 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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	6	4	1	2	3	5

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

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

Opgave 48 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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

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

Opgave 49 (4 %)

check

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

Hvad er resultatet af at anvende 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
7	8	9	4	6	5	1	3	2

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	4	6	5	1	3	2

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

Opgave 50 (4 %)

check

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

Hvad er resultatet af at anvende BUILD-MAX-HEAP på ovenstående array?

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

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

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
25	24	19	18	23	10	16	14	6	1	7	4	8

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
24	23	19	18	8	10	16	14	6	1	7	4

1	2	3	4	5	6	7	8	9	10	11	12	13
24	23	19	18	7	10	16	14	6	1		4	8

1	2	3	4	5	6	7	8	9	10	11	12
24	23	19	18	7	10	16	14	6	1	8	4

1	2	3	4	5	6	7	8	9	10	11	12
24	23	19	18	7	10	16	14	6	1	4	8

1	2	3	4	5	6	7	8	9	10	11	12
24	23	18	19	10	16	14	6	1	7	4	8

Opgave 52 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	23	8	16	18	9	4	3	7	10	13	15

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
24	16	23	8	10	18	9	4	3	7	15	13

1	2	3	4	5	6	7	8	9	10	11	12
24	23	15	16	18	9	4	3	7	10	13	8

1	2	3	4	5	6	7	8	9	10	11	12
24	16	23	8	10	18	9	4	3	7	13	15

1	2	3	4	5	6	7	8	9	10	11	12	13
24	16	23	8	10	18	9	4	3	7		13	15

1	2	3	4	5	6	7	8	9	10	11	12
24	16	23	8	15	18	9	4	3	7	10	13

Opgave 53 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	22	19	17	13	20	7	6	11	5	9	8

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
24	19	22	7	17	13	20	8	6	11	5	9

1	2	3	4	5	6	7	8	9	10	11	12	13
24	19	22	7	17	13	20		6	11	5	9	8

1	2	3	4	5	6	7	8	9	10	11	12
24	19	22	8	17	13	20	7	6	11	5	9

1	2	3	4	5	6	7	8	9	10	11	12
24	22	20	17	13	19	7	6	11	5	9	8

1	2	3	4	5	6	7	8	9	10	11	12
24	19	22	7	17	13	20	6	11	5	9	8

Opgave 54 (4 %)

check

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

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

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

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

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

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

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

Opgave 55 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	22	21	19	20	5	1	16	14	11	3	18

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
24	22	21	19	20	18	1	16	14	11	3	5

1	2	3	4	5	6	7	8	9	10	11	12
24	21	22	16	19	20	5	1	14	11	3	18

1	2	3	4	5	6	7	8	9	10	11	12
24	21	22	18	19	20	5	1	16	14	11	3

1	2	3	4	5	6	7	8	9	10	11	12	13
24	21	22	16	19	20	5	1		14	11	3	18

1	2	3	4	5	6	7	8	9	10	11	12
24	21	22	16	19	20	5	1	18	14	11	3

Opgave 56 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	18	22	19	13	4	7	9	17	12	6	10

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

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

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

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

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

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

Opgave 57 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
25	24	21	18	14	19	11	8	4	9	6	2	13

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
24	21	18	14	19	13	8	4	9	6	2	11

1	2	3	4	5	6	7	8	9	10	11	12
24	18	21	13	14	19	11	8	4	9	6	2

1	2	3	4	5	6	7	8	9	10	11	12
24	18	21	8	14	19	11	4	9	6	2	13

1	2	3	4	5	6	7	8	9	10	11	12
24	18	21	8	14	19	11	13	4	9	6	2

1	2	3	4	5	6	7	8	9	10	11	12	13
24	18	21	8	14	19	11		4	9	6	2	13

Opgave 58 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
23	22	18	21	15	16	13	6	9	11	14	5	12

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12	13
22	21	18	9	15	16	13	6		11	14	5	12

1	2	3	4	5	6	7	8	9	10	11	12
22	21	18	9	15	16	13	6	12	11	14	5

1	2	3	4	5	6	7	8	9	10	11	12
22	18	21	15	16	13	6	9	11	14	5	12

1	2	3	4	5	6	7	8	9	10	11	12
22	21	18	9	15	16	13	6	11	14	5	12

1	2	3	4	5	6	7	8	9	10	11	12
22	21	18	12	15	16	13	6	9	11	14	5

Opgave 59 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	22	19	10	13	20	8	7	5	1	6	9

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12
25	19	22	8	10	13	20	9	7	5	1	6

1	2	3	4	5	6	7	8	9	10	11	12
25	19	22	9	10	13	20	8	7	5	1	6

1	2	3	4	5	6	7	8	9	10	11	12
25	22	20	10	13	19	8	7	5	1	6	9

1	2	3	4	5	6	7	8	9	10	11	12
25	19	22	8	10	13	20	7	5	1	6	9

1	2	3	4	5	6	7	8	9	10	11	12	13
25	19	22	8	10	13	20		7	5	1	6	9

Opgave 60 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	22	13	23	19	8	11	12	2	10	4	15

Hvad er resultatet af at anvende HEAP-EXTRACT-MAX på ovenstående max-heap?

1	2	3	4	5	6	7	8	9	10	11	12	13
24	23	22	13	10	19	8	11	12	2		4	15

1	2	3	4	5	6	7	8	9	10	11	12
24	23	22	13	10	19	8	11	12	2	4	15

1	2	3	4	5	6	7	8	9	10	11	12
24	23	15	22	19	13	11	12	2	10	4	8

1	2	3	4	5	6	7	8	9	10	11	12
24	23	22	13	15	19	8	11	12	2	10	4

1	2	3	4	5	6	7	8	9	10	11	12
24	23	22	13	10	19	8	11	12	2	15	4

Partition

Opgave 61 (4 %)

check

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

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

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

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

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

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

Opgave 62 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	21	26	23	28	18	22	19	8	4	20	6	24	12	3

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	4	6	23	28	18	22	19	8	21	20	26	24	12	3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	4	6	8	12	15	18	19	20	21	22	23	24	26	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	4	6	8	18	19	20	21	22	23	26	28	24	12	3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	4	6	21	26	23	28	18	22	19	8	20	24	12	3

Opgave 63 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	17	8	23	29	2	18	26	22	20	25	11	1	9	28

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	8	9	11	17	18	20	21	22	23	25	26	28	29
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	17	1	8	23	29	2	18	26	22	20	25	11	9	28
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	17	1	2	8	11	18	20	22	23	25	26	29	9	28
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	17	1	23	29	2	18	26	22	20	25	11	8	9	28

Opgave 64 (4 %)

check

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

Angiv resultatet af at anvende $\text{PARTITION}(A, 4, 12)$ på ovenstående array A .

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

Opgave 65 (4 %)

check

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

Angiv resultatet af at anvende $\text{PARTITION}(A, 4, 12)$ på ovenstående array A .

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

Opgave 66 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	12	26	1	5	19	8	28	29	9	22	18	2	27	21

Angiv resultatet af at anvende PARTITION($A, 4, 13$) på ovenstående array A .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	12	26	1	2	19	8	28	29	9	22	18	5	27	21
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	12	26	1	2	5	8	9	18	19	22	28	29	27	21
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	12	26	1	2	5	19	8	28	29	9	22	18	27	21
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	5	8	9	12	18	19	21	22	26	27	28	29	30

Opgave 67 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	26	15	10	27	25	19	6	8	3	16	23	4	1	22

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
12	26	1	10	27	25	19	6	8	3	16	23	4	15	22
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	4	6	8	10	12	15	16	19	22	23	25	26	27
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	26	1	15	10	27	25	19	6	8	3	16	23	4	22
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	26	1	3	4	6	8	10	15	16	19	23	25	27	22

Opgave 68 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
19	24	1	11	23	6	18	21	22	10	3	2	25	13	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
19	24	1	11	6	10	3	2	13	23	18	21	22	25	12
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
19	24	1	2	3	6	10	11	13	18	21	22	23	25	12
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	6	10	11	12	13	18	19	21	22	23	24	25
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
19	24	1	11	6	10	3	2	13	23	18	21	25	22	12

Opgave 69 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	27	11	15	21	23	10	12	1	5	16	19	13	18	28

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
7	11	10	12	1	5	13	15	21	23	16	19	27	18	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	11	10	12	1	5	13	27	15	21	23	16	19	18	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	5	7	10	11	12	13	15	16	18	19	21	23	27	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	1	5	10	11	12	13	15	16	19	21	23	27	18	28

Opgave 70 (4 %)

check

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	5	19	13	28	22	15	29	30	17	2	18	12	14	10

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
11	5	2	12	13	14	15	17	18	19	22	28	29	30	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	10	11	12	13	14	15	17	18	19	22	28	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	5	13	2	12	14	19	28	22	15	29	30	17	18	10

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	5	13	2	12	14	15	29	30	17	19	18	28	22	10

Radix-sort

Opgave 71 (4 %)

check

2143 2102 0424 3324 1143 1224

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

2102 0424 3324 1224 2143 1143
 2102 2143 1143 0424 3324 1224
 0424 1143 1224 2102 2143 3324
 0424 1143 1224 2143 2102 3324
 2102 0424 1224 3324 1143 2143

Opgave 72 (4 %)

check

4010 3330 4410 1013 1430 1010

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

1010 1013 1430 3330 4010 4410
 4010 4410 1010 1013 3330 1430
 4010 4410 1010 3330 1430 1013
 1013 1010 1430 3330 4010 4410
 1010 4010 4410 1013 1430 3330

Opgave 73 (4 %)

check

2124 3404 2024 4324 1013 2013

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

1013 2013 3404 2124 2024 4324
 1013 2024 2013 2124 3404 4324
 1013 2013 2024 2124 3404 4324
 3404 1013 2013 2024 2124 4324
 3404 1013 2013 2124 2024 4324

Opgave 74 (4 %)

check

1143 4432 0234 1134 1432 4034

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

4432	1432	0234	1134	4034	1143
0234	1143	1134	1432	4034	4432
0234	1134	1143	1432	4034	4432
1432	4432	0234	1134	4034	1143
4432	1432	1143	0234	1134	4034

Opgave 75 (4 %)

check

2042 0041 4041 2241 4112 2212

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

4112	2212	0041	4041	2241	2042
0041	2042	2241	2212	4041	4112
0041	2042	2212	2241	4041	4112
0041	4041	2241	4112	2212	2042
2212	4112	0041	2241	4041	2042

Opgave 76 (4 %)

check

4040 2123 0430 0330 0423 4123

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

0430	0330	4040	2123	0423	4123
0330	0423	0430	2123	4040	4123
0330	0430	0423	2123	4040	4123
0423	2123	4123	0330	0430	4040
2123	0423	4123	0430	0330	4040

Opgave 77 (4 %)

check

3331 4143 2031 0031 4012 2012

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

4012	2012	3331	2031	0031	4143
0031	2031	2012	3331	4012	4143
0031	2012	2031	3331	4012	4143
2012	4012	0031	2031	3331	4143
3331	2031	0031	4012	2012	4143

Opgave 78 (4 %)

check

0321 4000 2121 2302 2300 1102

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

4000	2300	0321	2121	2302	1102
4000	2300	2302	1102	0321	2121
2300	4000	1102	2302	0321	2121
0321	1102	2121	2300	2302	4000
0321	1102	2121	2302	2300	4000

Opgave 79 (4 %)

check

4033 4313 2433 3120 4333 4320

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

4313	3120	4320	4033	2433	4333
4313	3120	4320	2433	4033	4333
2433	3120	4033	4313	4333	4320
3120	4320	4313	4033	2433	4333
2433	3120	4033	4313	4320	4333

Opgave 80 (4 %)

check

3241 2123 1023 0023 3233 0433

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

0023	0433	1023	2123	3241	3233
2123	1023	0023	3233	0433	3241
3241	2123	1023	0023	3233	0433
0023	0433	1023	2123	3233	3241
0023	1023	2123	0433	3233	3241

Lineær probing

Opgave 81 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11				5	16				3	14

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

Angiv positionerne de fem elementer 0, 2, 7, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 5, 11, 14 og 16).

[illegible]

Opgave 82 (4 %)

check

[illegible]

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

Angiv positionerne de fem elementer 1, 2, 3, 5 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 6, 15, 21 og 22).

[illegible]

Opgave 83 (4 %)

check

[illegible]

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

Angiv positionerne de fem elementer 1, 3, 4, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 6, 15, 17 og 22).

[illegible]

Opgave 84 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11	22	20							4	13

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

Angiv positionerne de fem elementer 0, 2, 3, 5 og 6 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 11, 13, 20 og 22).

[illegible]

Opgave 85 (4 %)

check

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

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

Angiv positionerne de fem elementer 2, 4, 6, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 3, 5, 14 og 21).

[illegible]

Opgave 86 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11	0					21			4	15

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, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 4, 11, 15 og 21).

[illegible]

Opgave 87 (4 %)

check

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

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

Angiv positionerne de fem elementer 4, 5, 6, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 2, 3, 14 og 16).

[illegible]

Opgave 88 (4 %)

check

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

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

Angiv positionerne de fem elementer 0, 2, 3, 5 og 6 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 7, 15, 16, 18 og 22).

[illegible]

Opgave 89 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
22	20	0							15	2

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

Angiv positionerne de fem elementer 4, 6, 7, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 2, 15, 20 og 22).

[illegible]

Opgave 90 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
		1			19			15	4	16

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

Angiv positionerne de fem elementer 2, 6, 7, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 4, 15, 16 og 19).

[illegible]

Kvadratisk probing

Opgave 91 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
		1	12			13			18	7

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

Angiv positionerne de fem elementer 0, 2, 3, 5 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 7, 12, 13 og 18).

[illegible]

Opgave 92 (4 %)

check

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

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 4k \bmod 11$ og $h(k, i) = (h'(k) + 2i + 3i^2) \bmod 11$.

Angiv positionerne de fem elementer 2, 3, 7, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 5, 10, 18 og 22).

[illegible]

Opgave 93 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
22	6					10		15	17	

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, 2, 4, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 10, 15, 17 og 22).

[illegible]

Opgave 94 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
	15	6			20		17			18

I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 3k \bmod 11$ og $h(k, i) = (h'(k) + 2i + 4i^2) \bmod 11$.

Angiv positionerne de fem elementer 0, 4, 7, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 15, 17, 18 og 20).

[illegible]

Opgave 95 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
0					12	10		17	11	

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

Angiv positionerne de fem elementer 1, 3, 4, 5 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 10, 11, 12 og 17).

[illegible]

Opgave 96 (4 %)

check

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

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

Angiv positionerne de fem elementer 1, 2, 5, 7 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 4, 11, 21 og 22).

[illegible]

Opgave 97 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
3	17					14	20		21	

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

Angiv positionerne de fem elementer 2, 4, 7, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 14, 17, 20 og 21).

[illegible]

Opgave 98 (4 %)

check

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

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

Angiv positionerne de fem elementer 0, 1, 2, 3 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 11, 18, 19 og 22).

[illegible]

Opgave 99 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11			18	13			20	22		

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 2, 5, 6, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 11, 13, 18, 20 og 22).

[illegible]

Opgave 100 (4 %)

check

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

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

Angiv positionerne de fem elementer 2, 4, 6, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 8, 13, 15 og 22).

[illegible]

Dobbelt hashing

Opgave 101 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
		19	12		14		6		3	

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 2, 5, 8, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 6, 12, 14 og 19).

[illegible]

Opgave 102 (4 %)

check

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

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 1, 2, 4, 6 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 7, 8, 9, 18 og 19).

[illegible]

Opgave 103 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
11	9			13			22			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, 7 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 9, 11, 13, 16 og 22).

[illegible]

Opgave 104 (4 %)

check

0	1	2	3	4	5	6	7	8	9	10
10							21	19	16	8

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 4k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 0, 1, 2, 3 og 4 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 10, 16, 19 og 21).

[illegible]

Opgave 105 (4 %)

check

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

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

Angiv positionerne de fem elementer 2, 3, 4, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 5, 6, 19 og 22).

[illegible]

Opgave 106 (4 %)

check

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

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 4k \bmod 11$ og $h_2(k) = 1 + (2k \bmod 10)$.

Angiv positionerne de fem elementer 0, 3, 4, 5 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 2, 7, 11 og 18).

[illegible]

Opgave 107 (4 %)

check

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

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

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 1, 3, 7, 11 og 14).

[illegible]

Opgave 108 (4 %)

check

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

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

Angiv positionerne de fem elementer 0, 1, 2, 6 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 8, 11, 12 og 22).

[illegible]

Opgave 109 (4 %)

check

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

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 3, 4, 5, 6 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 10, 12, 21 og 22).

[illegible]

Opgave 110 (4 %)

check

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

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

Angiv positionerne de fem elementer 3, 7, 8, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 1, 12, 18 og 22).

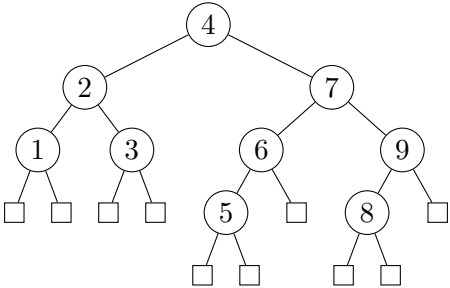
	0	1	2	3	4	5	6	7	8	9	10
INSERT(3)											110.1
INSERT(7)											110.2
INSERT(8)											110.3
INSERT(9)											110.4
INSERT(10)											110.5

Lovlige rød-sortede træer

Opgave 111 (4 %)

check

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

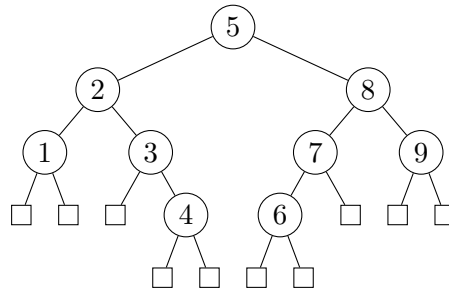


	Ja	Nej
1, 3, 5, 7, 8		111.1
5, 8		111.2
4, 5, 8		111.3
2, 5, 6, 8, 9		111.4
2, 5, 7, 8		111.5

Opgave 112 (4 %)

check

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



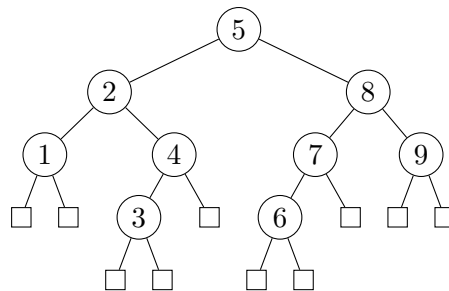
Ja Nej

- | | |
|---------------|-------|
| 2, 4, 6, 7, 9 | 112.1 |
| 1, 3, 4, 6, 8 | 112.2 |
| 4, 5, 6 | 112.3 |
| 2, 4, 6, 8 | 112.4 |
| 4, 6 | 112.5 |

Opgave 113 (4 %)

check

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



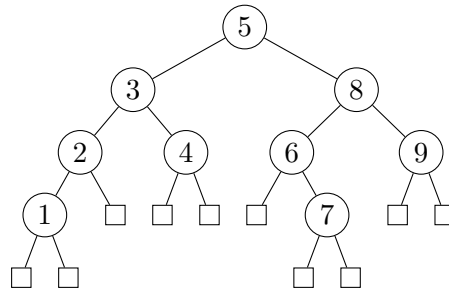
Ja Nej

- | | |
|---------------|-------|
| 3, 5, 6 | 113.1 |
| 2, 3, 6, 7, 9 | 113.2 |
| 2, 3, 6, 8 | 113.3 |
| 1, 3, 4, 6, 8 | 113.4 |
| 3, 6 | 113.5 |

Opgave 114 (4 %)

check

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



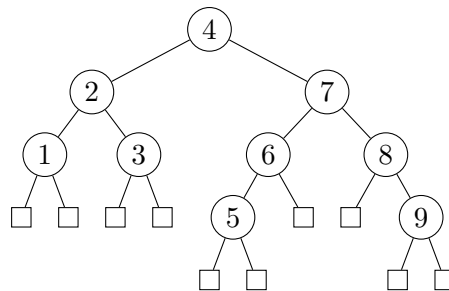
Ja Nej

- | | |
|---------------|-------|
| 1, 7 | 114.1 |
| 1, 5, 7 | 114.2 |
| 1, 3, 7, 8 | 114.3 |
| 1, 2, 4, 7, 8 | 114.4 |
| 1, 3, 6, 7, 9 | 114.5 |

Opgave 115 (4 %)

check

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



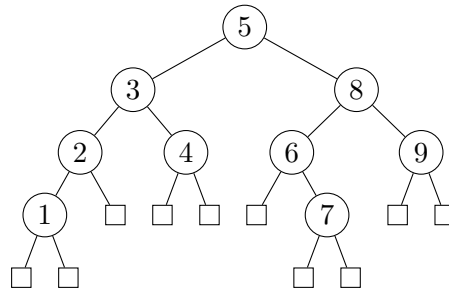
Ja Nej

- | | |
|------------------|-------|
| 4, 5, 9 | 115.1 |
| 1, 3, 5, 6, 8, 9 | 115.2 |
| 1, 3, 5, 7, 9 | 115.3 |
| 5, 9 | 115.4 |
| 2, 5, 7, 9 | 115.5 |

Opgave 116 (4 %)

check

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



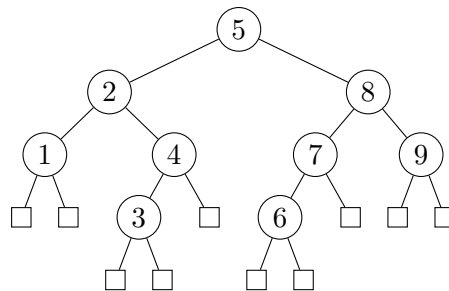
Ja Nej

- | | |
|---------------|-------|
| 1, 5, 7 | 116.1 |
| 1, 3, 7, 8 | 116.2 |
| 1, 2, 4, 7, 8 | 116.3 |
| 1, 7 | 116.4 |
| 1, 3, 6, 7, 9 | 116.5 |

Opgave 117 (4 %)

check

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



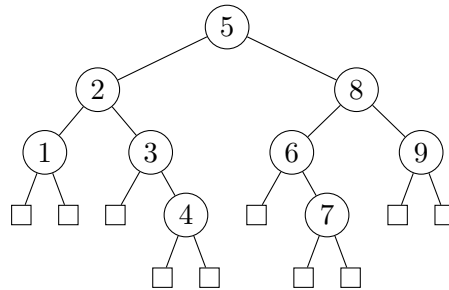
Ja Nej

- | | |
|---------------|-------|
| 3, 6 | 117.1 |
| 1, 3, 4, 6, 8 | 117.2 |
| 3, 5, 6 | 117.3 |
| 2, 3, 6, 8 | 117.4 |
| 2, 3, 6, 7, 9 | 117.5 |

Opgave 118 (4 %)

check

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



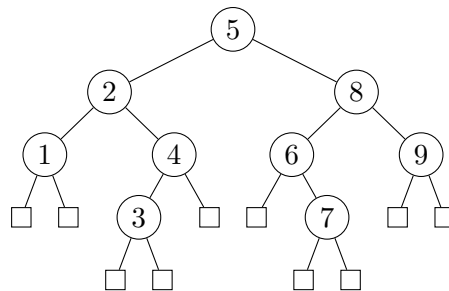
Ja Nej

4, 7	118.1
1, 3, 4, 7, 8	118.2
2, 4, 7, 8	118.3
2, 4, 6, 7, 9	118.4
4, 5, 7	118.5

Opgave 119 (4 %)

check

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



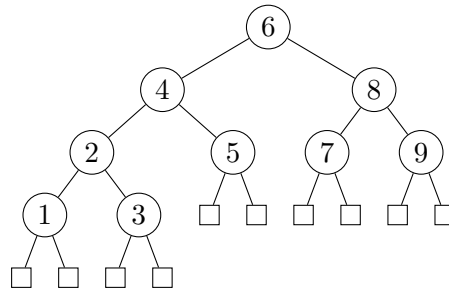
Ja Nej

3, 7	119.1
1, 3, 4, 7, 8	119.2
3, 5, 7	119.3
2, 3, 7, 8	119.4
2, 3, 6, 7, 9	119.5

Opgave 120 (4 %)

check

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



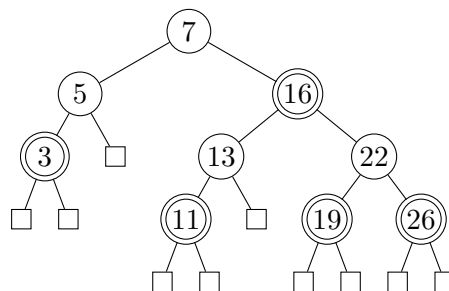
Ja Nej

- | | |
|------------------|-------|
| 1, 3, 4, 7, 9 | 120.1 |
| 1, 3, 6 | 120.2 |
| 1, 2, 3, 5, 7, 9 | 120.3 |
| 1, 3, 4, 8 | 120.4 |
| 1, 3 | 120.5 |

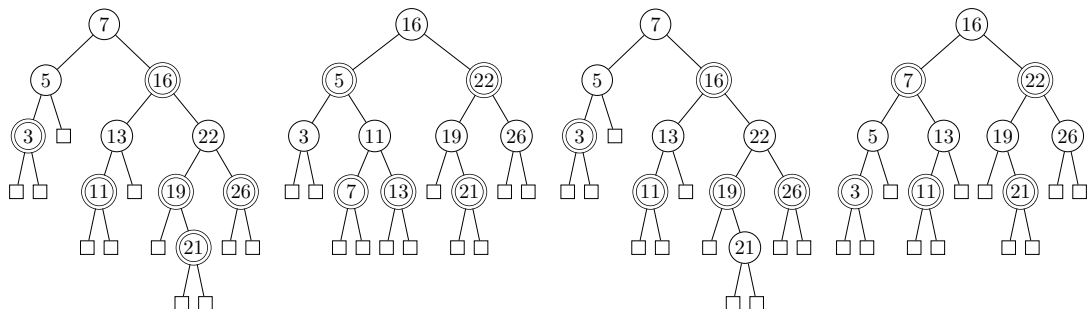
Rød-sort træ indsættelse

Opgave 121 (4 %)

check

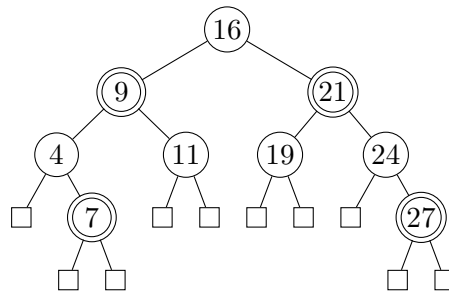


Angiv det resulterende rød-sort træ når man indsætter 21 i ovenstående rød-sort træ (dobbeltcirkler angiver røde knuder).

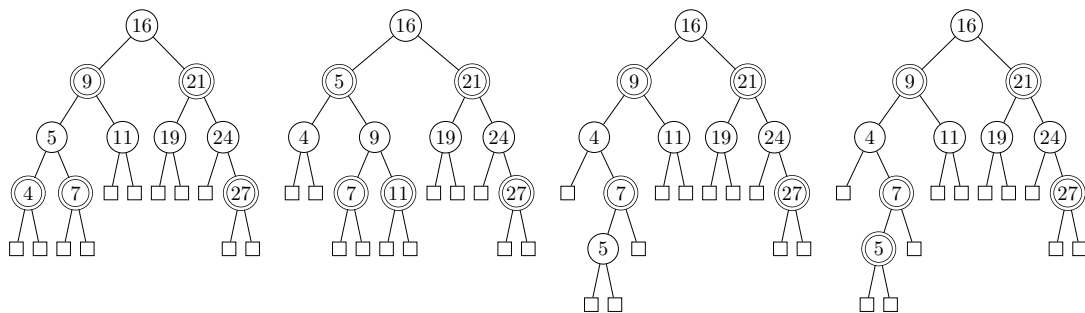


Opgave 122 (4 %)

check

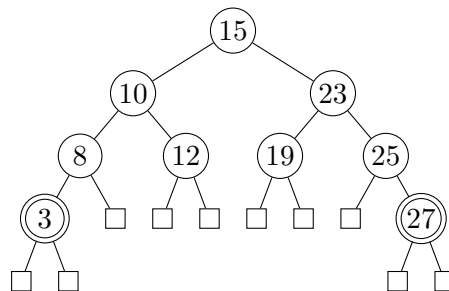


Angiv det resulterende rød-sortede træ når man indsætter 5 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

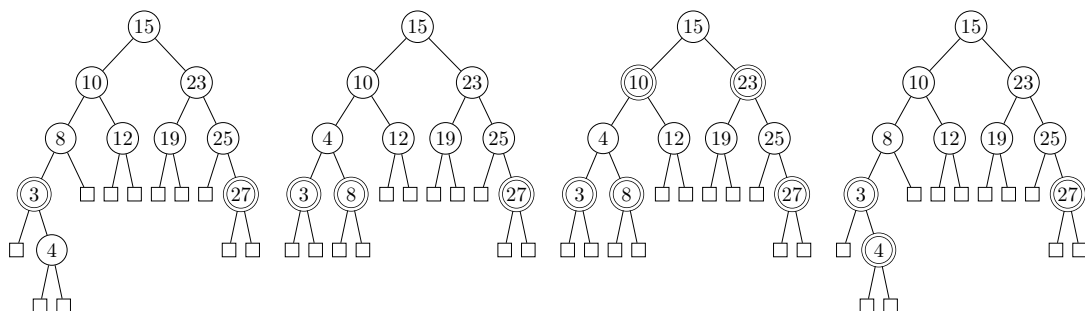


Opgave 123 (4 %)

check

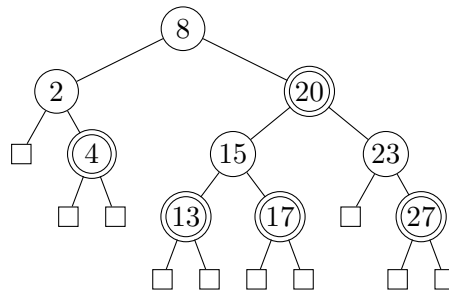


Angiv det resulterende rød-sortede træ når man indsætter 4 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

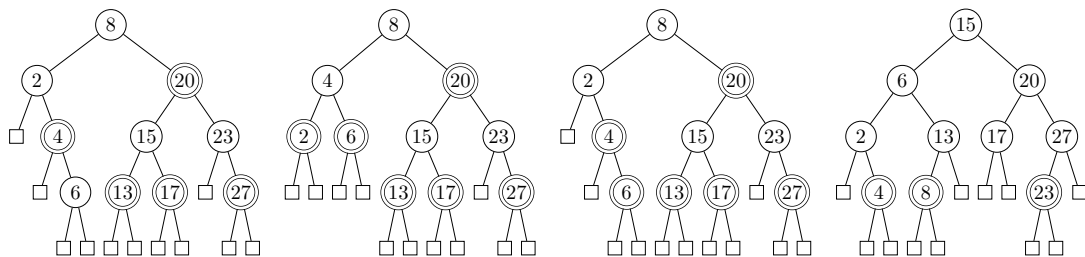


Opgave 124 (4 %)

check

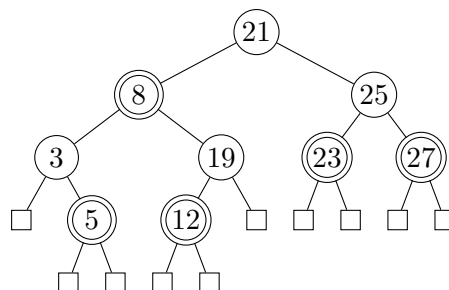


Angiv det resulterende rød-sortede træ når man indsætter 6 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

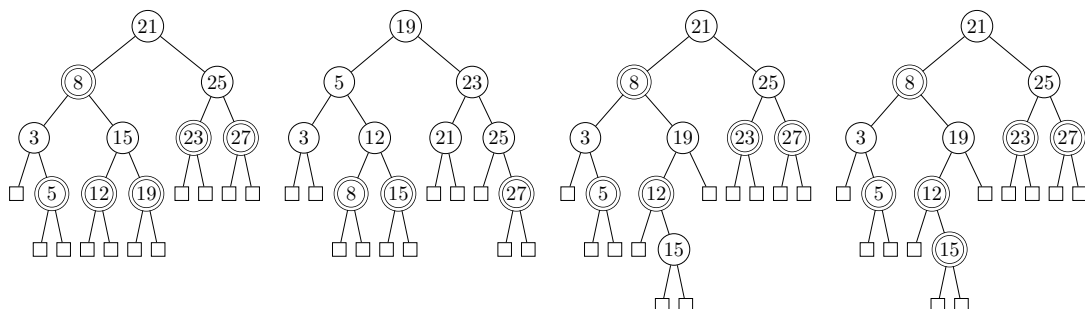


Opgave 125 (4 %)

check

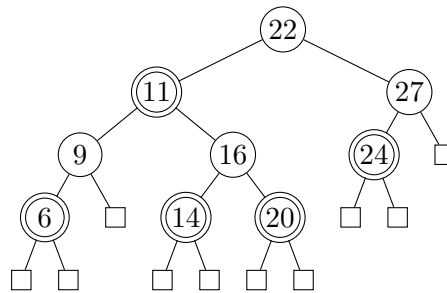


Angiv det resulterende rød-sortede træ når man indsætter 15 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

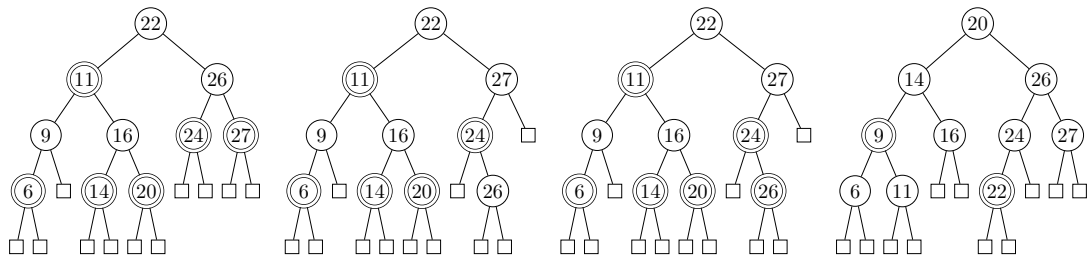


Opgave 126 (4 %)

check

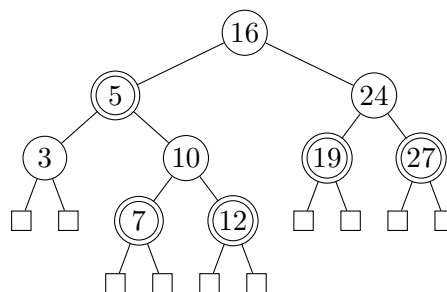


Angiv det resulterende rød-sortede træ når man indsætter 26 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

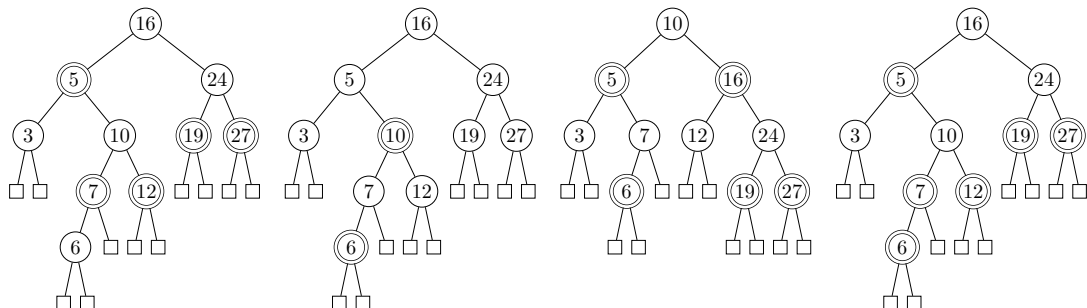


Opgave 127 (4 %)

check

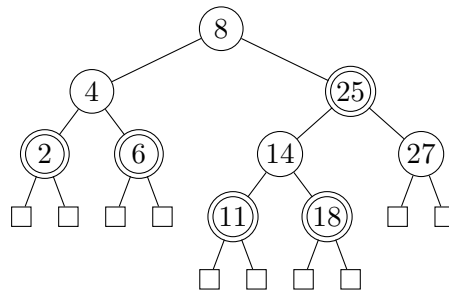


Angiv det resulterende rød-sortede træ når man indsætter 6 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

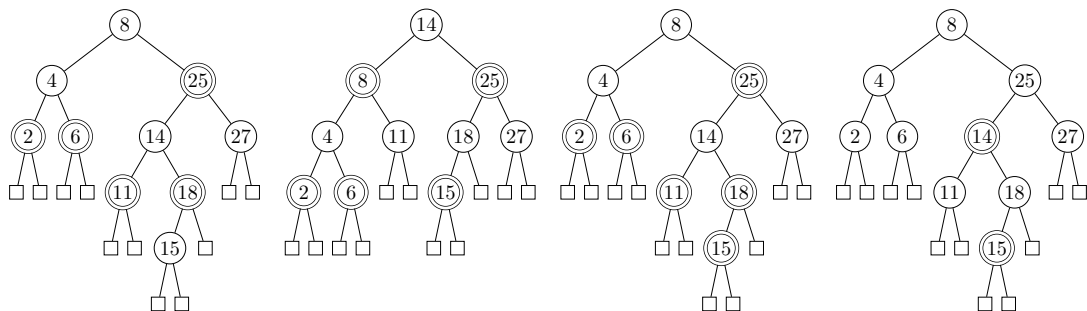


Opgave 128 (4 %)

check

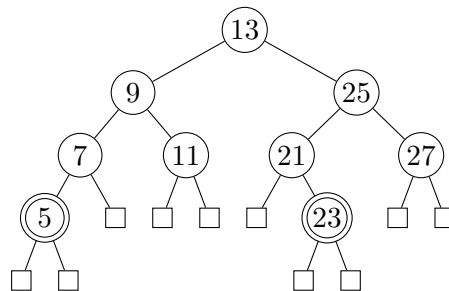


Angiv det resulterende rød-sortte træ når man indsætter 15 i ovenstående rød-sortte træ (dobbeltcirkler angiver røde knuder).

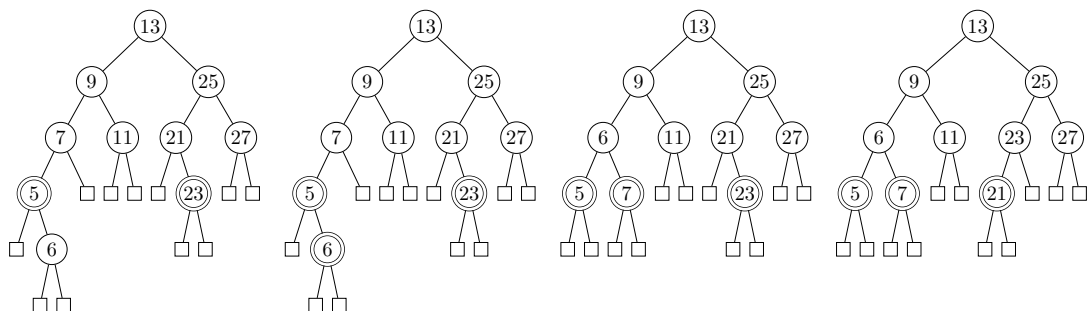


Opgave 129 (4 %)

check

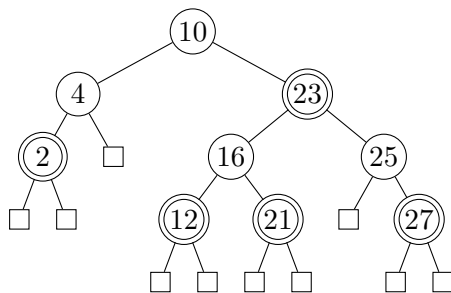


Angiv det resulterende rød-sortte træ når man indsætter 6 i ovenstående rød-sortte træ (dobbeltcirkler angiver røde knuder).

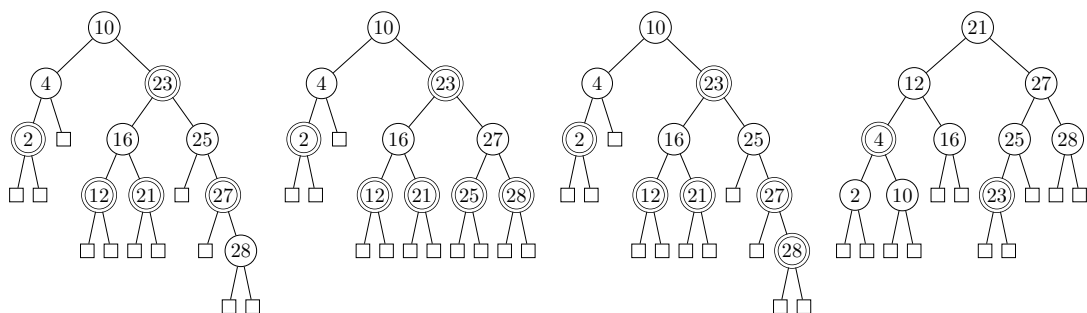


Opgave 130 (4 %)

check



Angiv det resulterende rød-sortede træ når man indsætter 28 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



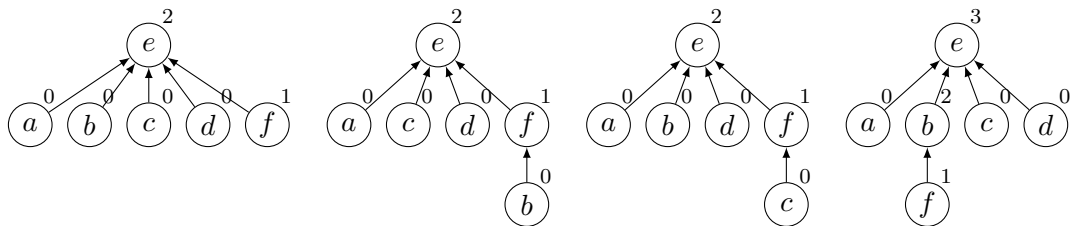
Union-find

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

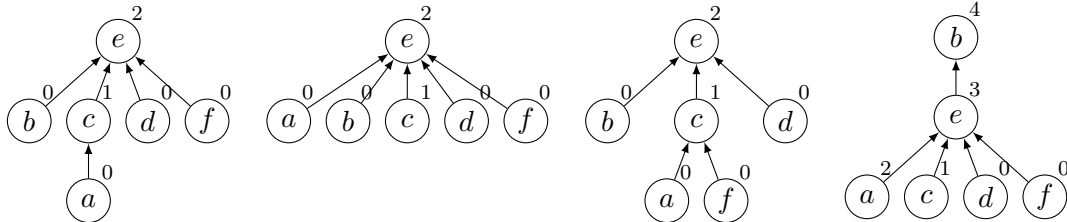


Opgave 132 (4 %)

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

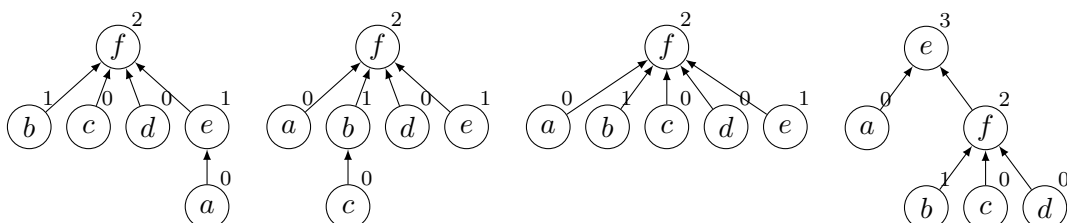


Opgave 133 (4 %)

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

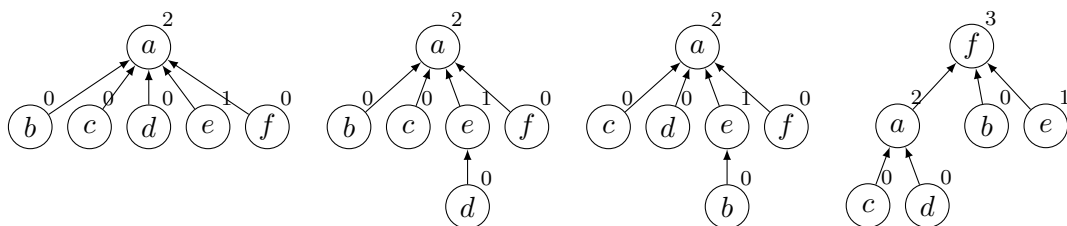


Opgave 134 (4 %)

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

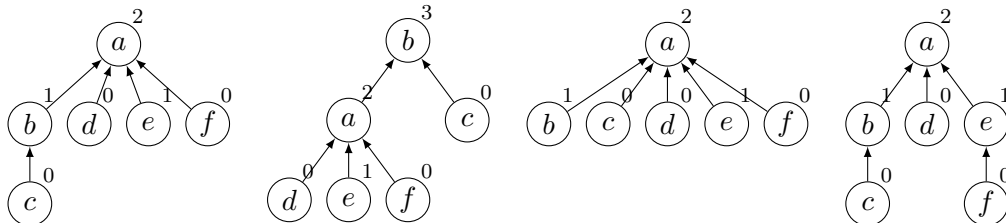


Opgave 135 (4 %)

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

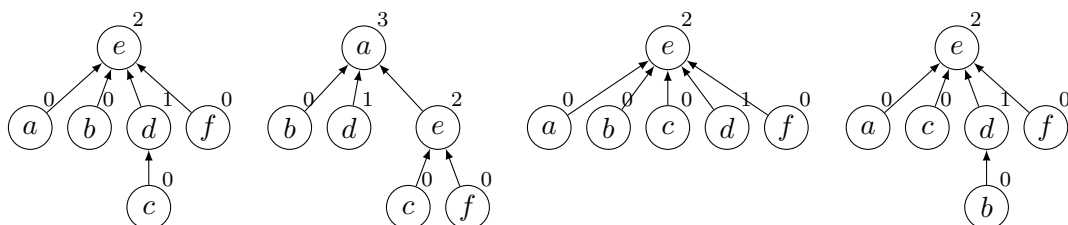


Opgave 136 (4 %)

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

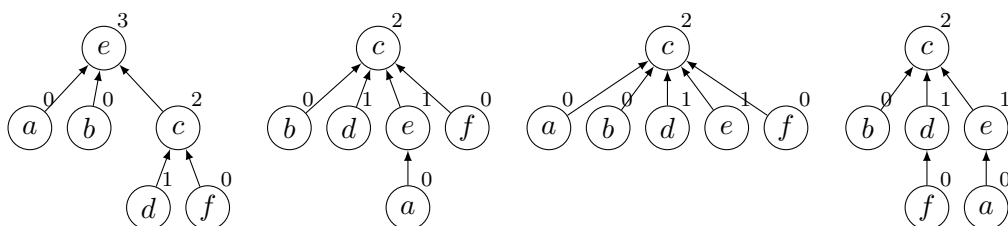


Opgave 137 (4 %)

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

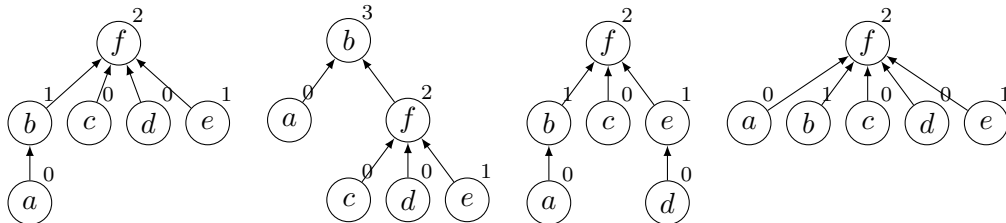


Opgave 138 (4 %)

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

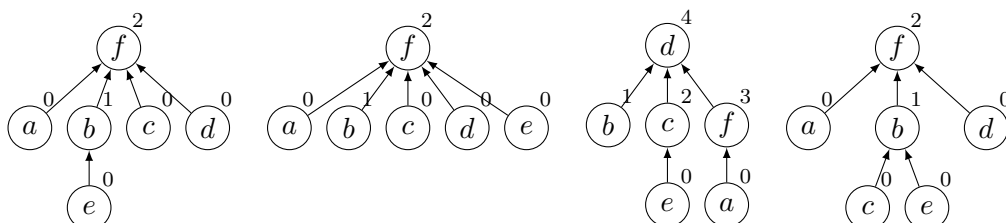


Opgave 139 (4 %)

check

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

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

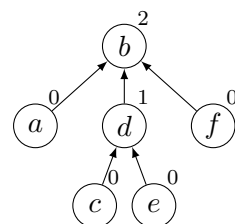
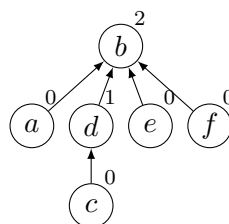
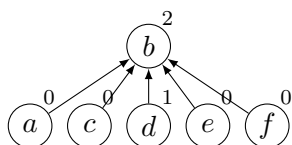
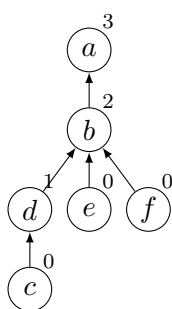


Opgave 140 (4 %)

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



Huffman koder

Opgave 141 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	20	80	10	40	70	40	30

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **a**, **d** og **f**?

	1	2	3	4	5	6
a						141.1
d						141.2
f						141.3

Filen indeholder $20 + 80 + 10 + 40 + 70 + 40 + 30 = 290$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

750	790	800	810	830	840
					141.4

Opgave 142 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	60	30	50	80	80	40	30

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **b**, **d** og **g**?

	1	2	3	4	5	6
b						142.1
d						142.2
g						142.3

Filen indeholder $60 + 30 + 50 + 80 + 80 + 40 + 30 = 370$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

1010	1020	1030	1040	1050	1060
					142.4

Opgave 143 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	40	10	30	20	40	70	80

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **c**, **d** og **e**?

	1	2	3	4	5	6	
c							143.1
d							143.2
e							143.3

Filen indeholder $40 + 10 + 30 + 20 + 40 + 70 + 80 = 290$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

730 740 750 780 790 800

143.4

Opgave 144 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	40	20	70	60	80	50	10

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **d**, **e** og **g**?

	1	2	3	4	5	6	
d							144.1
e							144.2
g							144.3

Filen indeholder $40 + 20 + 70 + 60 + 80 + 50 + 10 = 330$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

830 850 860 870 900 910

144.4

Opgave 145 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	70	10	50	50	10	30	20

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **a**, **c** og **f**?

	1	2	3	4	5	6
a						145.1
c						145.2
f						145.3

Filen indeholder $70 + 10 + 50 + 50 + 10 + 30 + 20 = 240$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

590 600 610 620 630 640

145.4

Opgave 146 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	40	80	60	20	10	40	30

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **c**, **e** og **g**?

	1	2	3	4	5	6
c						146.1
e						146.2
g						146.3

Filen indeholder $40 + 80 + 60 + 20 + 10 + 40 + 30 = 280$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

720 730 760 770 780 790

146.4

Opgave 147 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	70	40	50	30	20	20	30

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **c**, **d** og **f**?

	1	2	3	4	5	6
c						147.1
d						147.2
f						147.3

Filen indeholder $70 + 40 + 50 + 30 + 20 + 20 + 30 = 260$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

660 670 680 700 710 720

147.4

Opgave 148 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	80	50	50	60	10	40	20

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **b**, **c** og **g**?

	1	2	3	4	5	6
b						148.1
c						148.2
g						148.3

Filen indeholder $80 + 50 + 50 + 60 + 10 + 40 + 20 = 310$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

780 800 810 820 830 850

148.4

Opgave 149 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	50	20	70	30	30	40	20

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **b**, **d** og **e**?

	1	2	3	4	5	6
b						149.1
d						149.2
e						149.3

Filen indeholder $50 + 20 + 70 + 30 + 30 + 40 + 20 = 260$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

700 710 720 730 740 750

149.4

Opgave 150 (4 %)

check

Bogstav	a	b	c	d	e	f	g
Frekvens	40	60	60	10	50	50	20

Antag vi har en fil med ovenstående bogstaver og frekvenser. Hvis man konstruerer et Huffman træ for disse frekvenser, hvor lange er koderne så for **b**, **e** og **f**?

	1	2	3	4	5	6
b						150.1
e						150.2
f						150.3

Filen indeholder $40 + 60 + 60 + 10 + 50 + 50 + 20 = 290$ bogstaver. Hvor mange bits vil en Huffman kodning af strengen bruge?

720 730 740 760 770 780

150.4

Rekursionsligninger

Opgave 151 (4 %)

check

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

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

$$T(n) = T(n-1) + \log n \quad 151.1$$

$$T(n) = 4 \cdot T(n/5) + n^3 \quad 151.2$$

$$T(n) = 4 \cdot T(n/2) + n^2 \quad 151.3$$

$$T(n) = 3 \cdot T(n/5) + n \quad 151.4$$

$$T(n) = T(n-1) + n^2 \quad 151.5$$

Opgave 152 (4 %)

check

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

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

$$T(n) = T(n-1) + \log n \quad 152.1$$

$$T(n) = 4 \cdot T(n/5) + n \quad 152.2$$

$$T(n) = T(n/3) + 3 \quad 152.3$$

$$T(n) = 2 \cdot T(n/2) + n \quad 152.4$$

$$T(n) = 3 \cdot T(n/9) + 3 \quad 152.5$$

Opgave 153 (4 %)

check

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

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

$$T(n) = 2 \cdot T(n/4) + 2 \quad 153.1$$

$$T(n) = 5 \cdot T(n/5) + n \quad 153.2$$

$$T(n) = 3 \cdot T(n/9) + 1 \quad 153.3$$

$$T(n) = 8 \cdot T(n/2) + 2 \quad 153.4$$

$$T(n) = 4 \cdot T(n/2) + n^2 \quad 153.5$$

Opgave 154 (4 %)

check

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

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

$$T(n) = 2 \cdot T(n/5) + n \quad 154.1$$

$$T(n) = 4 \cdot T(n/2) + 1 \quad 154.2$$

$$T(n) = 4 \cdot T(n/4) + n \quad 154.3$$

$$T(n) = T(n/4) + 3 \quad 154.4$$

$$T(n) = T(n-1) + n \quad 154.5$$

Opgave 155 (4 %)

check

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

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

$$T(n) = 2 \cdot T(n/4) + 1 \quad 155.1$$

$$T(n) = 5 \cdot T(n/5) + n \quad 155.2$$

$$T(n) = 8 \cdot T(n/2) + 1 \quad 155.3$$

$$T(n) = T(n-1) + 2 \quad 155.4$$

$$T(n) = T(n-1) + n \quad 155.5$$

Opgave 156 (4 %)

check

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

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

$$T(n) = 4 \cdot T(n/2) + n^2 \quad 156.1$$

$$T(n) = 9 \cdot T(n/3) + 3 \quad 156.2$$

$$T(n) = 4 \cdot T(n/2) + 2 \quad 156.3$$

$$T(n) = 2 \cdot T(n/4) + 2 \quad 156.4$$

$$T(n) = 8 \cdot T(n/2) + 1 \quad 156.5$$

Opgave 157 (4 %)

check

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

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

$$T(n) = 8 \cdot T(n/2) + 2 \quad 157.1$$

$$T(n) = T(n/3) + 5 \quad 157.2$$

$$T(n) = T(n-1) + n \quad 157.3$$

$$T(n) = 2 \cdot T(n/4) + n \quad 157.4$$

$$T(n) = 3 \cdot T(n/4) + n^3 \quad 157.5$$

Opgave 158 (4 %)

check

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

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

$$T(n) = 9 \cdot T(n/3) + n^2 \quad 158.1$$

$$T(n) = 4 \cdot T(n/2) + 2 \quad 158.2$$

$$T(n) = T(n/4) + 3 \quad 158.3$$

$$T(n) = 3 \cdot T(n/9) + 1 \quad 158.4$$

$$T(n) = 3 \cdot T(n/5) + n \quad 158.5$$

Opgave 159 (4 %)

check

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

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

$$T(n) = 3 \cdot T(n/9) + 3 \quad 159.1$$

$$T(n) = T(n-1) + \log n \quad 159.2$$

$$T(n) = 2 \cdot T(n/4) + 2 \quad 159.3$$

$$T(n) = T(n-1) + 3 \quad 159.4$$

$$T(n) = T(n/3) + 5 \quad 159.5$$

Opgave 160 (4 %)

check

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

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

$$T(n) = 9 \cdot T(n/3) + 3 \quad 160.1$$

$$T(n) = 4 \cdot T(n/5) + n^2 \quad 160.2$$

$$T(n) = 4 \cdot T(n/2) + n^2 \quad 160.3$$

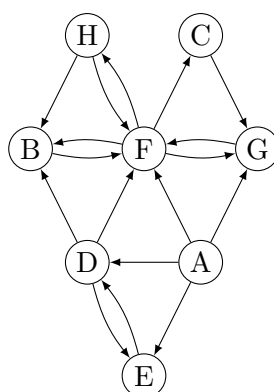
$$T(n) = 3 \cdot T(n/9) + 2 \quad 160.4$$

$$T(n) = 2 \cdot T(n/4) + n^3 \quad 160.5$$

BFS

Opgave 161 (4 %)

check

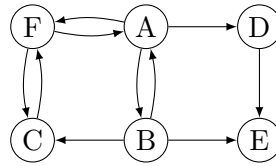


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ADEFGBCH ADEFGBHC AGDEFBCH ADBFCGHE

Opgave 162 (4 %)

check

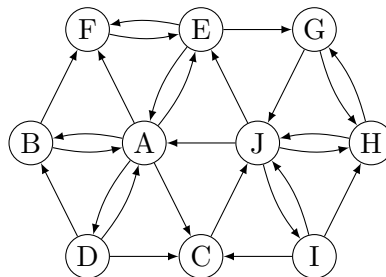


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ABCFED ABDFCE ABDFEC AFDBCE

Opgave 163 (4 %)

check

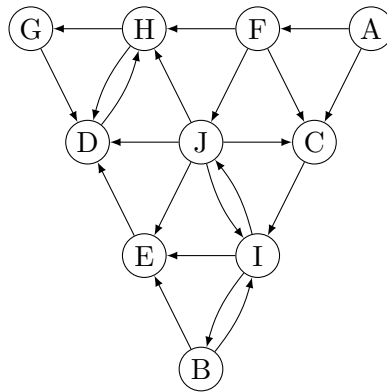


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ABFEGHJICD ABCDEFJGHI AEBFDCGJHI ABCDEFJGHI

Opgave 164 (4 %)

check

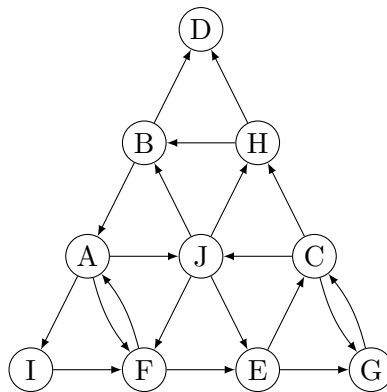


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ACIBEDHGGJF ACFIHJBEGD ACFIHJEBGD ACFIHJBEDG

Opgave 165 (4 %)

check

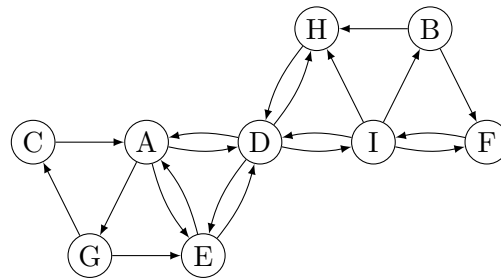


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

AFIJEHBHCGD AJFIHBEDCG AFIJEHBHCGD AFECGHBDJI

Opgave 166 (4 %)

check

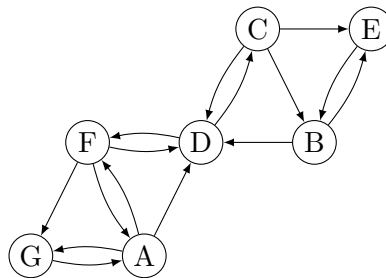


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ADEGHICBF ADEHIBFGC ADEGHICFB ADEGIHCFB

Opgave 167 (4 %)

check

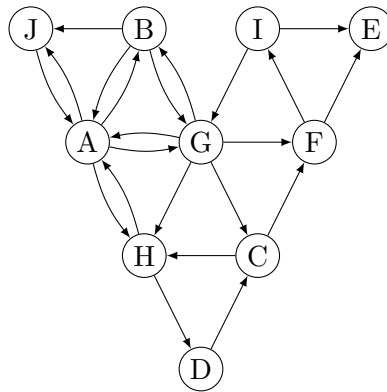


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ADFGCBE ADGFCEB ADFGCEB ADCBEFG

Opgave 168 (4 %)

check

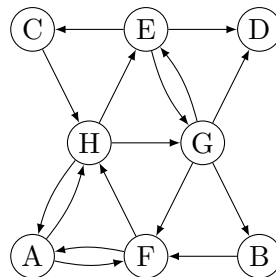


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ABGCFEIH DJ ABGHJCFDIE ABGHJCFDEI ABHJGDCFEI

Op­gave 169 (4 %)

check

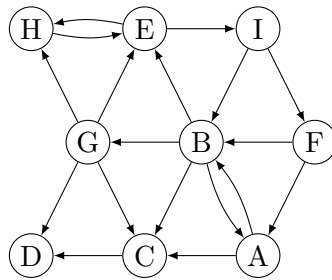


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

AFHECDGB AHFGEBDC AFHEGDCB AFHEGCDB

Opgave 170 (4 %)

check



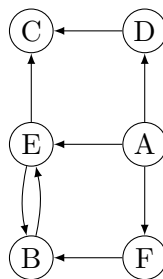
For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved alfabetisk sorterede incidenslister.

ABCDEHIFG ACBDGEHIF ABCEGDHIF ABCEGDIHF

Lovlige BFS træer

Opgave 171 (4 %)

check



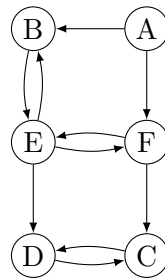
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

- | | |
|-------------------------------|-------|
| (A,D) (A,F) (B,E) (E,C) (F,B) | 171.1 |
| (A,D) (A,E) (A,F) (E,B) (E,C) | 171.2 |
| (A,D) (A,E) (A,F) (D,C) (E,B) | 171.3 |
| (A,D) (A,F) (B,E) (D,C) (F,B) | 171.4 |
| (A,D) (A,E) (A,F) (E,C) (F,B) | 171.5 |

Opgave 172 (4 %)

check



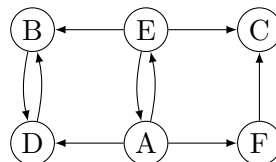
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,B) (A,F) (E,D) (F,C) (F,E)	172.1
(A,B) (B,E) (D,C) (E,D) (E,F)	172.2
(A,B) (A,F) (D,C) (E,D) (F,E)	172.3
(A,F) (E,B) (E,D) (F,C) (F,E)	172.4
(A,B) (A,F) (C,D) (F,C) (F,E)	172.5

Opgave 173 (4 %)

check



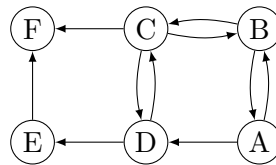
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,D) (A,E) (A,F) (D,B) (F,C)	173.1
(A,E) (A,F) (B,D) (E,B) (E,C)	173.2
(A,D) (A,E) (A,F) (E,B) (F,C)	173.3
(A,D) (A,E) (A,F) (E,B) (E,C)	173.4
(A,D) (A,E) (A,F) (D,B) (E,C)	173.5

Opgave 174 (4 %)

check



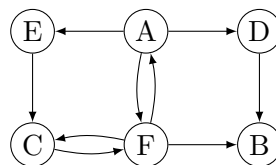
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

- | | |
|-------------------------------|-------|
| (A,B) (A,D) (C,F) (D,C) (D,E) | 174.1 |
| (A,D) (C,B) (D,C) (D,E) (E,F) | 174.2 |
| (A,B) (A,D) (D,C) (D,E) (E,F) | 174.3 |
| (A,B) (A,D) (B,C) (C,F) (D,E) | 174.4 |
| (A,B) (A,D) (B,C) (D,E) (E,F) | 174.5 |

Opgave 175 (4 %)

check



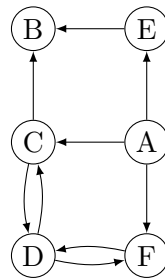
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

- | | |
|-------------------------------|-------|
| (A,D) (A,E) (A,F) (D,B) (F,C) | 175.1 |
| (A,D) (A,E) (A,F) (D,B) (E,C) | 175.2 |
| (A,D) (A,E) (A,F) (E,C) (F,B) | 175.3 |
| (A,D) (A,E) (C,F) (E,C) (F,B) | 175.4 |
| (A,D) (A,E) (A,F) (F,B) (F,C) | 175.5 |

Opgave 176 (4 %)

check



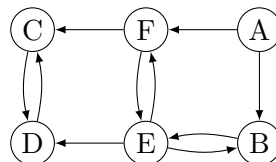
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,C) (A,E) (A,F) (C,D) (E,B)	176.1
(A,C) (A,E) (A,F) (E,B) (F,D)	176.2
(A,E) (A,F) (C,B) (D,C) (F,D)	176.3
(A,C) (A,E) (A,F) (C,B) (C,D)	176.4
(A,C) (A,E) (A,F) (C,B) (F,D)	176.5

Opgave 177 (4 %)

check



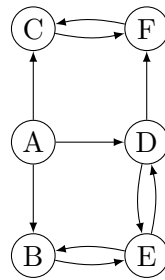
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,B) (A,F) (E,D) (F,C) (F,E)	177.1
(A,B) (B,E) (D,C) (E,D) (E,F)	177.2
(A,B) (A,F) (B,E) (E,D) (F,C)	177.3
(A,F) (C,D) (E,B) (F,C) (F,E)	177.4
(A,B) (A,F) (B,E) (D,C) (E,D)	177.5

Opgave 178 (4 %)

check



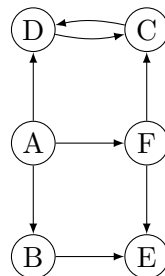
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,D) (D,E) (D,F) (E,B) (F,C)	178.1
(A,B) (A,C) (A,D) (C,F) (D,E)	178.2
(A,B) (A,C) (A,D) (B,E) (C,F)	178.3
(A,B) (A,C) (B,E) (C,F) (E,D)	178.4
(A,B) (A,C) (A,D) (D,E) (D,F)	178.5

Opgave 179 (4 %)

check



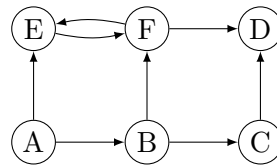
Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en passende ordning af grafens incidenslister.

Ja Nej

(A,B) (A,F) (C,D) (F,C) (F,E)	179.1
(A,B) (A,D) (A,F) (B,E) (F,C)	179.2
(A,B) (A,D) (A,F) (F,C) (F,E)	179.3
(A,B) (A,D) (A,F) (B,E) (D,C)	179.4
(A,B) (A,D) (A,F) (D,C) (F,E)	179.5

Opgave 180 (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 passende ordning af grafens incidenslister.

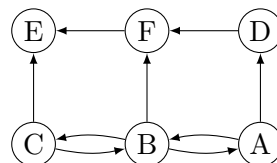
Ja Nej

(A,B) (B,C) (B,F) (F,D) (F,E)	180.1
(A,B) (A,E) (B,C) (C,D) (E,F)	180.2
(A,B) (A,E) (B,C) (B,F) (C,D)	180.3
(A,B) (A,E) (B,C) (E,F) (F,D)	180.4
(A,B) (A,E) (B,C) (B,F) (F,D)	180.5

DFS

Opgave 181 (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**.

ABCEFD ADBFCE ABDCFE ADFEBC

181.1

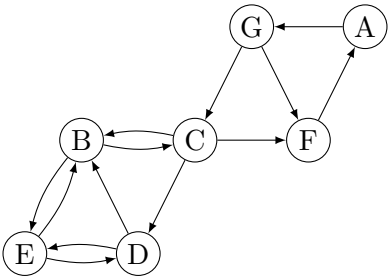
Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

Tree edge Back edge Cross edge Forward edge

(A, D)	181.2
(C, B)	181.3
(F, E)	181.4
(B, A)	181.5

Opgave 182 (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**.

AGCFBED AGCBEDF AGCFBDE AGFCBED

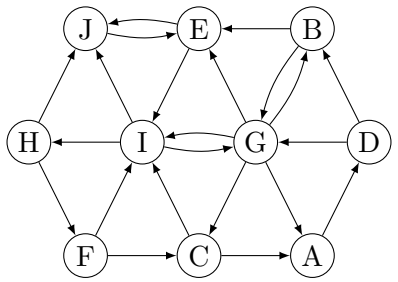
182.1

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

Tree edge	Back edge	Cross edge	Forward edge	
(C, D)				182.2
(D, E)				182.3
(D, B)				182.4
(C, B)				182.5

Opgave 183 (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**.

ADBGECIJHF ADBEIHJFGC ADBEIGCHFJ ADGBEJIHFC

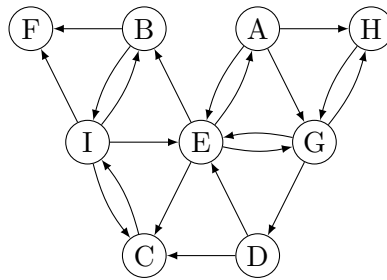
183.1

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

	Tree edge	Back edge	Cross edge	Forward edge	
(F, C)					183.2
(E, J)					183.3
(F, I)					183.4
(B, E)					183.5

Opgave 184 (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**.

FCIBDHGEA HDGCIFBEA FBICDHGEA IFDCBHGEA

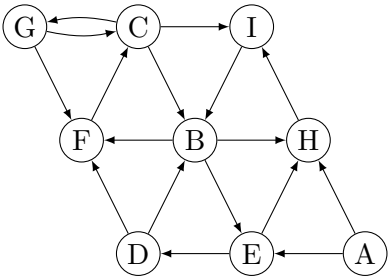
184.1

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

	Tree edge	Back edge	Cross edge	Forward edge	
(D, E)					184.2
(A, H)					184.3
(A, E)					184.4
(I, F)					184.5

Opgave 185 (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**.

AEDBHF CIG AEDBHIFCG AEDBF C G I H AEHDIBFCG

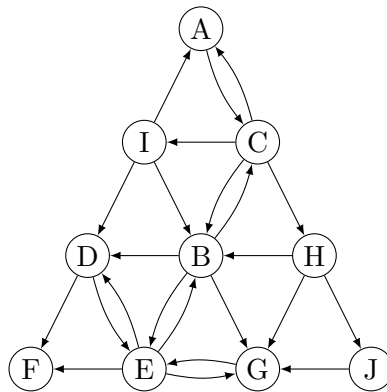
185.1

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

	Tree edge	Back edge	Cross edge	Forward edge	
(D, F)					185.2
(I, B)					185.3
(B, H)					185.4
(H, I)					185.5

Opgave 186 (4 %)

check



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

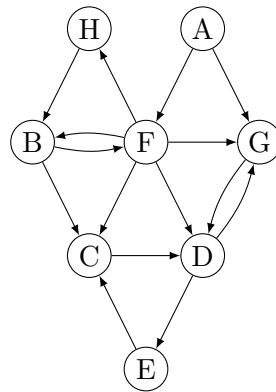
186.1

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

	Tree edge	Back edge	Cross edge	Forward edge
(D, F)				186.2
(D, E)				186.3
(I, A)				186.4
(I, D)				186.5

Opgave 187 (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**.

EGDCBHFA EHDCBGFA HGEDCBFA GEDCBHFA

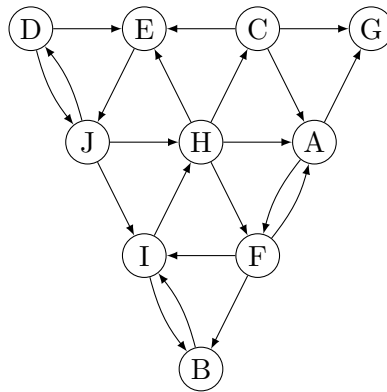
187.1

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

	Tree edge	Back edge	Cross edge	Forward edge	
(G, D)					187.2
(H, B)					187.3
(D, G)					187.4
(A, G)					187.5

Opgave 188 (4 %)

check



Betrakt 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**.

AFGBIHCEJD AFBHICGEJD AFBHCEJDG AGFIHCEJDB

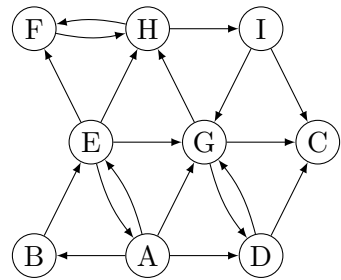
188.1

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

	Tree edge	Back edge	Cross edge	Forward edge
(H, C)				188.2
(D, J)				188.3
(F, I)				188.4
(I, B)				188.5

Opgave 189 (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**.

ABEFHICGD ABDEGCFHI ABEFHIGDC AEFHIGCDB

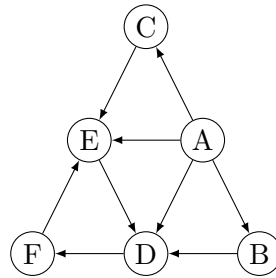
189.1

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

	Tree edge	Back edge	Cross edge	Forward edge	
(E, H)					189.2
(G, C)					189.3
(E, F)					189.4
(H, F)					189.5

Opgave 190 (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**.

CEFDBA EFDBCA FDEBCA FEDCBA

190.1

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

Tree edge Back edge Cross edge Forward edge

(F, E) 190.2

(E, D) 190.3

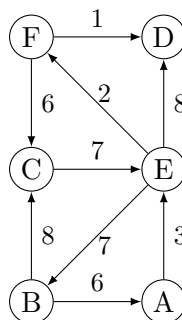
(C, E) 190.4

(A, E) 190.5

Dijkstras algoritme

Opgave 191 (4 %)

check

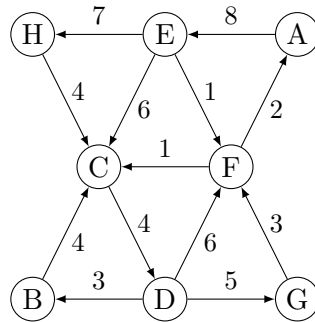


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AEBCDF AEBDFC AEFDCB AEFDBC

Opgave 192 (4 %)

check

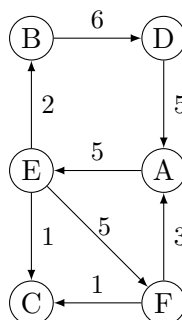


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AECDDBFGH AEFCDDBGH AECFHDBG AEFCDHGB

Opgave 193 (4 %)

check

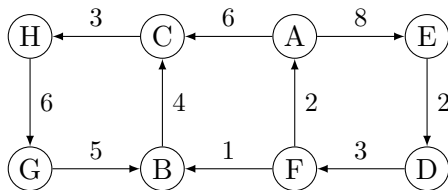


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AEBCFD AECBFD AECBDF AEBDCF

Opgave 194 (4 %)

check

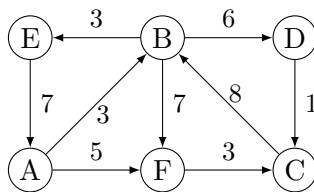


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

ACEHGBDF ACEHDGFB ACEHDFBG ACHGBEDF

Opgave 195 (4 %)

check

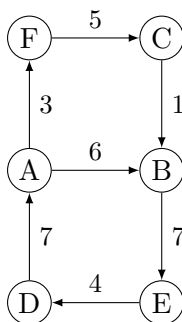


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

ABFCED ABDCEF ABFECD ABFDEC

Opgave 196 (4 %)

check

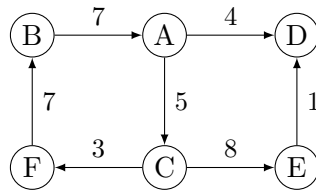


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AFBCED AFBEDC ABFECD ABEDFC

Opgave 197 (4 %)

check

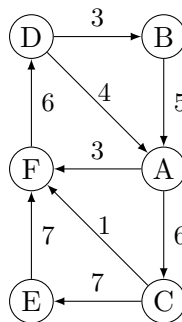


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

ADCFBE ADCFEB ACDEFB ACEDFB

Opgave 198 (4 %)

check

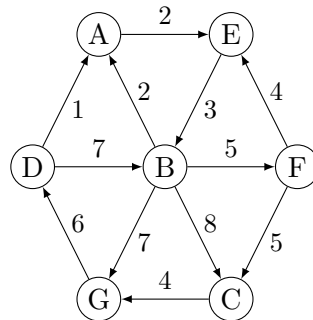


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

ACEFDB AFCEDB ACFEDB AFCDBE

Opgave 199 (4 %)

check

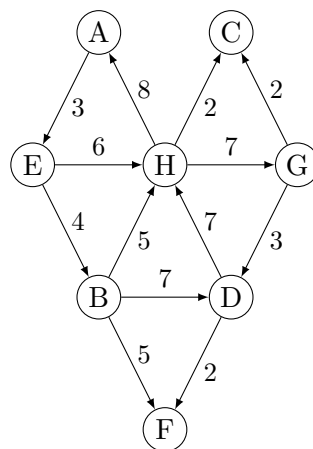


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AEBCFGD AEBCGDF AEBFCGD AEBFGCD

Opgave 200 (4 %)

check



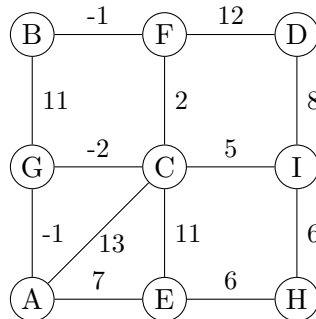
Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

AEBHCGDF AEBHCFDG AEBDFHCG AEBHDFCG

Prims algoritme

Opgave 201 (4 %)

check

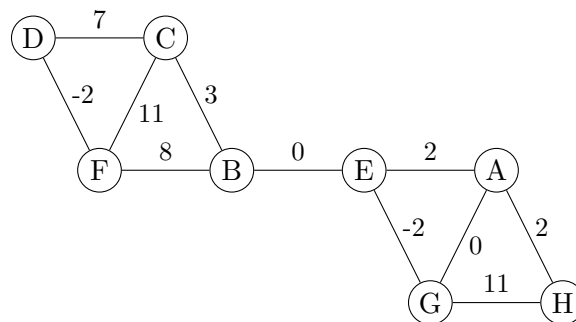


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AGCFBIHED AGCFBDIHE AGCFBIHDE AGCFBIEHD

Opgave 202 (4 %)

check

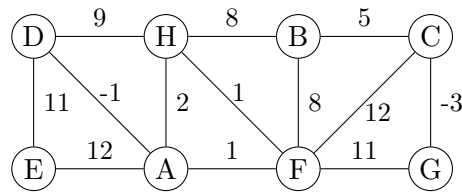


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AGEBHCDF AGEBCHFD AGEBCDFH AGEBCHDF

Opgave 203 (4 %)

check

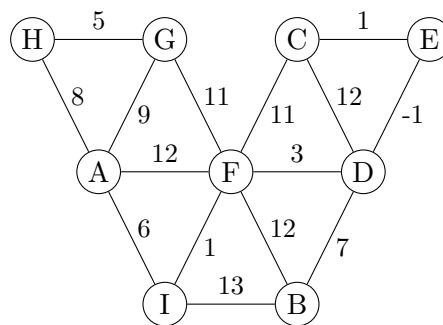


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

ADFHBEGC ADFHBCGE ADFHBECG ADHFBCGE

Opgave 204 (4 %)

check

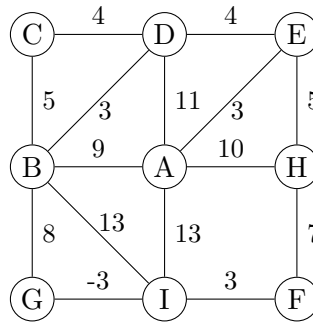


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AIFDECBHG AIFHGDECB AIFDECBGH AIFHDECGB

Opgave 205 (4 %)

check

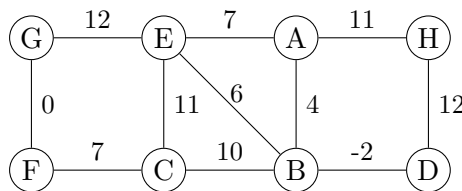


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AEDBCGIFH AEDHBCIGF AEDBCHFIG AEDHBCFIG

Opgave 206 (4 %)

check

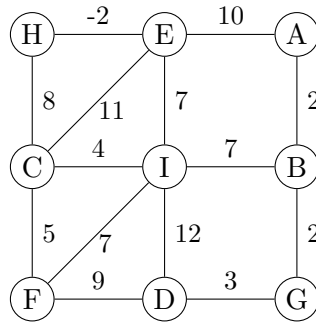


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

ABDEHCGF ABDHECFG ABDEHCFG ABDECFGH

Opgave 207 (4 %)

check

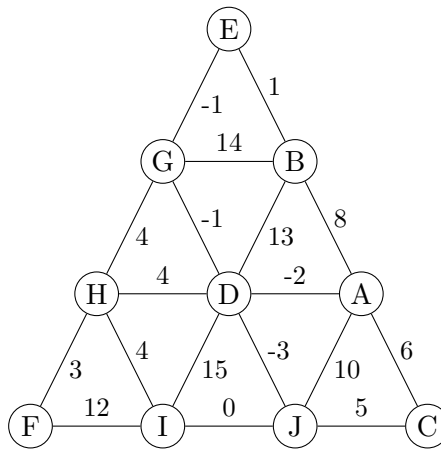


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

ABGDFCIEH ABGDIEHCF ABGDICFEH ABGDICEHF

Opgave 208 (4 %)

check

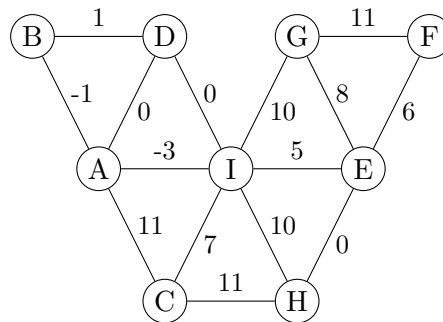


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

ADJIHFGEB C ADJGEIBHFC ADJIGEBCHF ADJIGEBHCF

Opgave 209 (4 %)

check

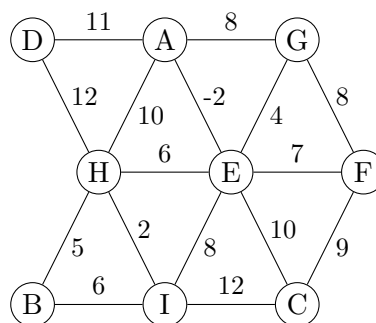


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AIBDEHCFG AIDBEHCGF AIDBEHCFG AIBDEHFCG

Opgave 210 (4 %)

check



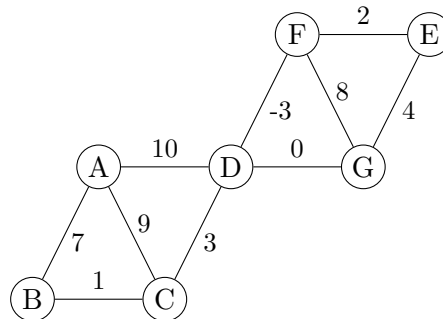
Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AEGHFIBDC AEGFCIHBD AEGHIBFCD AEGHFICBD

Kruskals algoritme

Opgave 211 (4 %)

check

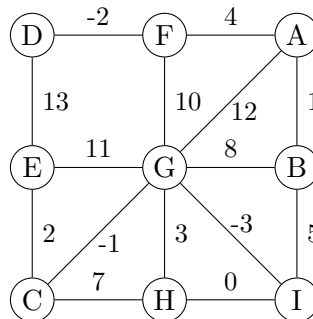


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(B, C) (A, B) (E, G) (E, F) (D, G)

Opgave 212 (4 %)

check

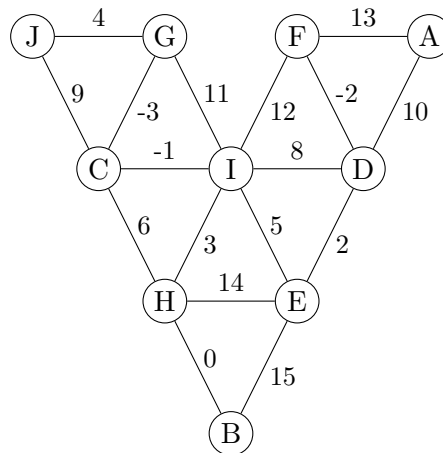


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(B, I) (C, H) (C, E) (H, I) (G, I)

Opgave 213 (4 %)

check

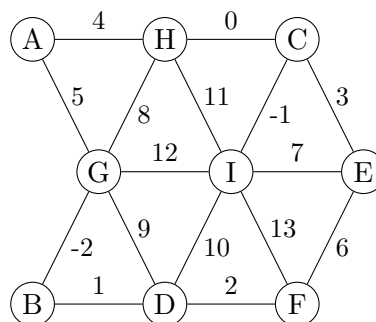


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(B, H) (B, E) (A, D) (C, I) (G, J)

Opgave 214 (4 %)

check

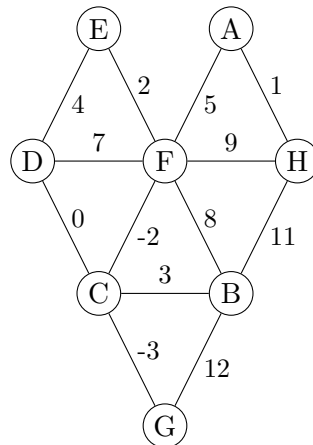


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(B, G) (A, G) (C, E) (A, H) (D, F)

Opgave 215 (4 %)

check

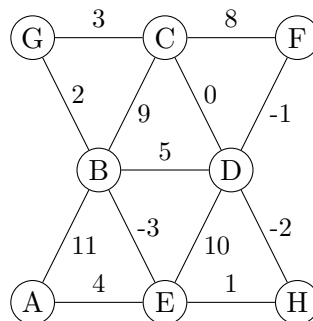


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

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

Opgave 216 (4 %)

check

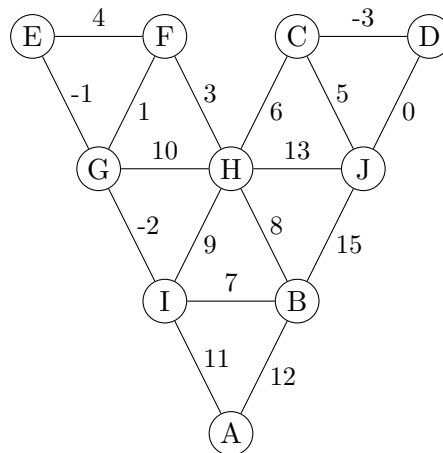


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(D,F) (C,D) (A,E) (B,G) (B,E)

Opgave 217 (4 %)

check

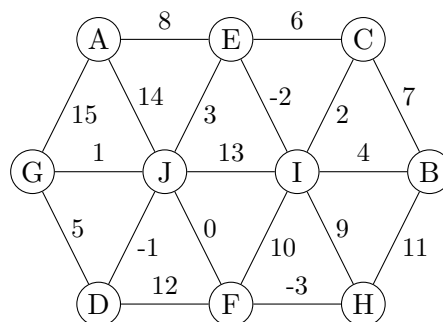


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(B,I) (C,H) (D,J) (B,J) (A,I)

Opgave 218 (4 %)

check

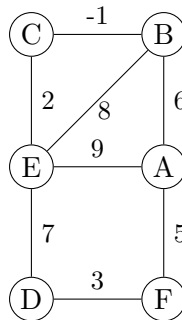


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(D,G) (B,I) (C,I) (A,E) (G,J)

Opgave 219 (4 %)

check

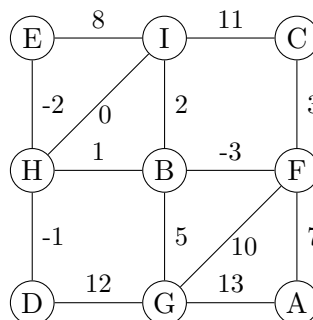


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

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

Opgave 220 (4 %)

check



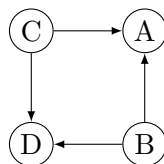
Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det minimum udspændende træ.

(C, F) (B, G) (B, H) (D, G) (A, F)

Topologisk sortering

Opgave 221 (4 %)

check

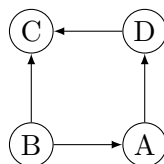


Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
B C D A		221.1
C B A D		221.2
C D A B		221.3
B C A D		221.4
D B A C		221.5

Opgave 222 (4 %)

check

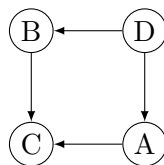


Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A B D C		222.1
C A D B		222.2
B D A C		222.3
B A D C		222.4
B A C D		222.5

Opgave 223 (4 %)

check



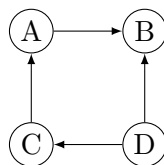
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

DCAB	223.1
DABC	223.2
DBAC	223.3
CABD	223.4
ADBC	223.5

Opgave 224 (4 %)

check



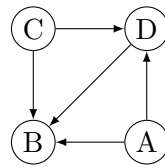
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

DCBA	224.1
DCAB	224.2
DACB	224.3
BCAD	224.4
CDAB	224.5

Opgave 225 (4 %)

check



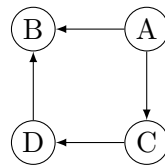
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

A C D B	225.1
B C D A	225.2
C A B D	225.3
C A D B	225.4
C D A B	225.5

Opgave 226 (4 %)

check



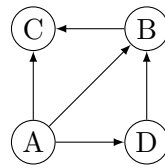
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

A C D B	226.1
A D C B	226.2
A B D C	226.3
D C A B	226.4
C A D B	226.5

Opgave 227 (4 %)

check



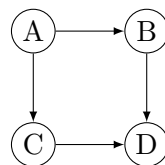
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

A D C B	227.1
A D B C	227.2
C D B A	227.3
A C B D	227.4
D A B C	227.5

Opgave 228 (4 %)

check



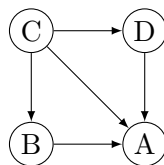
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

B A C D	228.1
C B A D	228.2
A C B D	228.3
A D B C	228.4
A B C D	228.5

Opgave 229 (4 %)

check



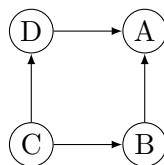
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

DBCA	229.1
CBAD	229.2
ABDC	229.3
CBDA	229.4
CDBA	229.5

Opgave 230 (4 %)

check



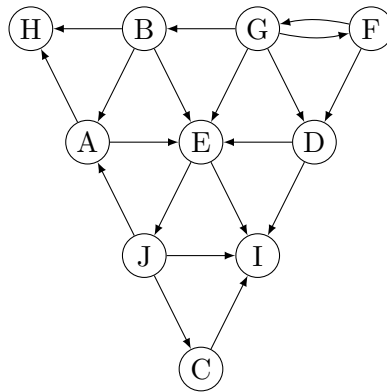
Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

Ja Nej

CDBA	230.1
CBDA	230.2
CDAB	230.3
CBAD	230.4
DBCA	230.5

Opgave 234 (4 %)

check

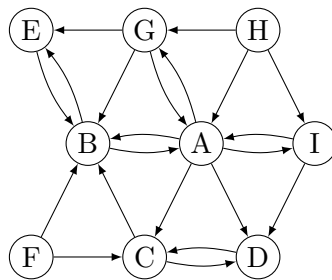


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9 10

Opgave 235 (4 %)

check

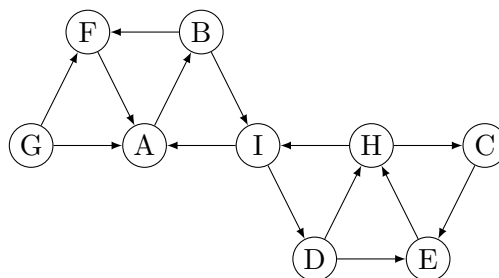


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Opgave 236 (4 %)

check

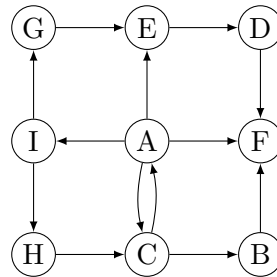


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Opgave 237 (4 %)

check

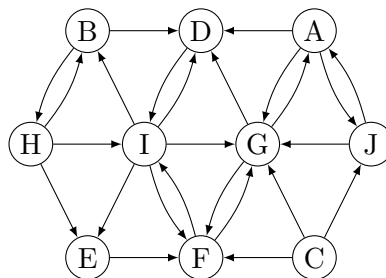


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Opgave 238 (4 %)

check

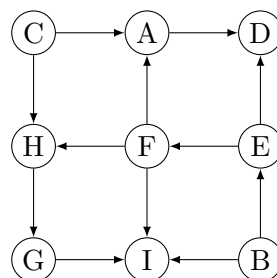


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9 10

Opgave 239 (4 %)

check

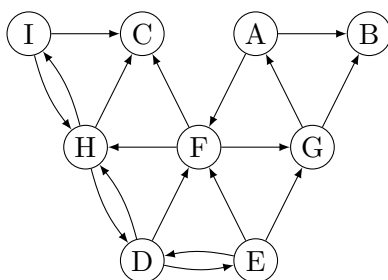


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Opgave 240 (4 %)

check



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

Løkke opgaver

Opgave 241

check

Algorithm loop1(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq i$ $j = 2 * j$ $i = 2 * i$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq n$ $j = 2 * j$ $i = 2 * i$	Algorithm loop3(n) $i = 1$ while $i \leq n$ $j = i$ while $j \leq n$ $j = 2 * j$ $i = 2 * i$
Algorithm loop4(n) $i = 1$ while $i \leq n$ $j = i$ while $j > 0$ $j = \lfloor j/2 \rfloor$ $i = i + i$	Algorithm loop5(n) $i = n$ while $i > 0$ $j = i$ while $j > 0$ $j = \lfloor j/2 \rfloor$ $i = \lfloor i/2 \rfloor$	Algorithm loop6(n) $s = 1$ for $i = 1$ to n $j = 1$ while $j \leq s$ $j = j + 1$ $s = 2 * s$
Algorithm loop7(n) $s = 1$ for $i = 1$ to n $j = s$ while $j > 0$ $s = s + 1$ $j = j - 1$	Algorithm loop8(n) $i = 1$ $p = 1$ while $p \leq n$ $i = i + 1$ $p = p * i$	Algorithm loop9(n) $i = 1$ $j = n$ while $i < j$ $i = 2 * i$ $j = j + n$

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

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

loop1	241.1
loop2	241.2
loop3	241.3
loop4	241.4
loop5	241.5
loop6	241.6
loop7	241.7
loop8	241.8
loop9	241.9

Opgave 242

check

Algorithm loop1(n) $i = 1$ $j = n$ while $i \leq j$ $i = 4 * i$ $j = 2 * j$	Algorithm loop2(n) $i = 1$ $j = n$ while $i \leq j$ $i = i * 2$ $j = \lfloor j/2 \rfloor$	Algorithm loop3(n) $i = 1$ while $i * i \leq n$ $i = i + i$
Algorithm loop4(n) $i = 1$ while $i \leq n$ $i = 2 * i$	Algorithm loop5(n) $i = 1$ while $i \leq n$ $i = 3 * i$	Algorithm loop6(n) $i = 1$ while $i \leq n$ $i = i + i$
Algorithm loop7(n) $i = 1$ while $i \leq n * n$ $i = 2 * i$	Algorithm loop8(n) $i = 1$ while $i \leq n * n$ $i = 3 * i$	Algorithm loop9(n) $i = 1$ $j = n * n$ while $i \leq j$ $i = 2 * i$ $j = j - 1$

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

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

loop1	242.1
loop2	242.2
loop3	242.3
loop4	242.4
loop5	242.5
loop6	242.6
loop7	242.7
loop8	242.8
loop9	242.9

Opgave 243

check

Algorithm loop1(n) $i = n$ while $i > 0$ if i ulige then $i = i - 1$ else $i = i/2$	Algorithm loop2(n) $i = n$ while $i \leq n * n$ $i = 2 * i$	Algorithm loop3(n) $s = n$ while $s > 0$ $s = \lfloor s/2 \rfloor$
Algorithm loop4(n) $i = 2$ while $i \leq n$ $i = i * i$	Algorithm loop5(n) $s = 2$ while $s \leq n$ $s = s * s$	Algorithm loop6(n) $i = 0$ $s = 0$ $q = 0$ while $q \leq n$ $i = i + 1$ $s = s + i$ $q = q + s$
Algorithm loop7(n) $i = 0$ $s = 0$ while $s \leq n$ $i = i + 1$ $s = s + i$	Algorithm loop8(n) $i = 1$ $j = 1$ $s = 0$ while $s \leq n$ while $j \leq s$ $j = 2 * j$ $s = s + i$ $i = i + 1$	Algorithm loop9(n) $j = n$ $i = 1$ while $j \geq 0$ $j = j - i$ $i = i + 1$

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

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

loop1	243.1
loop2	243.2
loop3	243.3
loop4	243.4
loop5	243.5
loop6	243.6
loop7	243.7
loop8	243.8
loop9	243.9

Opgave 244

check

Algorithm loop1(n) $s = 0$ $i = 1$ while $s \leq n$ $s = s + i$ $i = i + 1$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $j = 1$ $k = 1$ while $k \leq n$ $j = j + 1$ $k = k + j$ $i = 2 * i$	Algorithm loop3(n) for $i = 1$ to n $j = i$ while $j \leq n$ $j = 2 * j$
Algorithm loop4(n) $i = 0$ $j = n$ while $i \leq j$ $i = i + 1$ $j = j - 1$	Algorithm loop5(n) $i = 0$ $j = n$ while $i < j$ $i = i + 2$ $j = j + 1$	Algorithm loop6(n) $i = 1$ $j = 0$ while $i \leq n$ $i = i + i$ while $j < i$ $j = j + 1$
Algorithm loop7(n) $i = 1$ $j = 1$ while $i \leq n$ while $j \leq i$ $j = j + 1$ $i = 2 * i$	Algorithm loop8(n) $i = 1$ $s = 0$ while $s \leq n$ $j = 1$ while $j \leq i$ $j = j + 1$ $s = s + i$ $i = i + 1$	Algorithm loop9(n) $i = 1$ $s = 1$ while $s \leq n * n$ $i = i + 1$ $s = s + i$

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

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

loop1	244.1
loop2	244.2
loop3	244.3
loop4	244.4
loop5	244.5
loop6	244.6
loop7	244.7
loop8	244.8
loop9	244.9

Opgave 245

check

Algorithm loop1(n) $i = 1$ while $i \leq n$ $j = 0$ while $j \leq i$ $j = j + 1$ $i = 2 * i$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $j = 0$ while $j \leq n$ $j = j + i$ $i = 2 * i$	Algorithm loop3(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq i$ $j = j + 1$ $i = 2 * i$
Algorithm loop4(n) $i = n$ while $i > 0$ $i = i - 1$	Algorithm loop5(n) $i = n$ while $i \geq 1$ $j = i$ while $j \leq n$ $j = 2 * j$ $i = i - 1$	Algorithm loop6(n) $s = 0$ $i = 1$ while $i * i \leq n$ for $j = 1$ to i $s = s + 1$ $i = i + 1$
Algorithm loop7(n) $s = 0$ $i = n$ while $i > 1$ for $j = 1$ to i $s = s + 1$ $i = \lfloor i/2 \rfloor$	Algorithm loop8(n) $s = 1$ for $i = 1$ to n $s = s + 1$	Algorithm loop9(n) $s = 1$ for $i = n$ to 1 step -1 $s = s + 1$

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

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

loop1	245.1
loop2	245.2
loop3	245.3
loop4	245.4
loop5	245.5
loop6	245.6
loop7	245.7
loop8	245.8
loop9	245.9

Opgave 246

check

Algorithm loop1(n) $s = 1$ $i = 1$ while $i \leq n$ for $j = 1$ to i $s = s + 1$ $i = 2 * i$	Algorithm loop2(n) $s = 1$ while $s \leq n$ $s = s + 1$	Algorithm loop3(n) for $i = 1$ to n $j = 0$ while $j \leq n$ $j = j + i$
Algorithm loop4(n) for $i = 1$ to n $j = 1$ while $j \leq i$ $j = 2 * j$	Algorithm loop5(n) for $i = 1$ to n $j = 1$ while $j \leq n$ $j = 2 * j$	Algorithm loop6(n) for $i = 1$ to n $j = i$ while $j > 1$ $j = \lfloor j/2 \rfloor$
Algorithm loop7(n) $i = 0$ while $i \leq n$ $j = i$ while $j > 0$ $j = \lfloor j/2 \rfloor$ $i = i + 1$	Algorithm loop8(n) $i = 1$ $j = 1$ $s = 0$ while $i \leq n$ if $i = j$ then for $k = 1$ to n $s = s + 1$ $j = 2 * j$ $i = i + 1$	Algorithm loop9(n) $i = 1$ $s = 0$ while $i \leq n$ for $j = i$ to n $s = s + 1$ $i = i + i$

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

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

loop1	246.1
loop2	246.2
loop3	246.3
loop4	246.4
loop5	246.5
loop6	246.6
loop7	246.7
loop8	246.8
loop9	246.9

Opgave 247

check

Algorithm loop1(n) $i = 1$ while $i \leq n$ $j = 0$ while $j \leq n$ $j = j + 1$ $i = 2 * i$	Algorithm loop2(n) $i = 1$ while $i \leq n$ $j = 1$ while $j \leq i$ $j = 2 * j$ $i = i + 1$	Algorithm loop3(n) $i = 1$ while $i \leq n$ $j = i$ while $j \leq n$ $j = j + 1$ $i = 2 * i$
Algorithm loop4(n) $i = 1$ while $i \leq n$ $j = n$ while $j > 1$ $j = j - 1$ $i = 2 * i$	Algorithm loop5(n) $s = 0$ $i = n$ while $i > 1$ for $j = 1$ to n $s = s + 1$ $i = \lfloor i/2 \rfloor$	Algorithm loop6(n) for $i = 0$ to n $j = 0$ $s = 0$ while $s \leq i$ $j = j + 1$ $s = s + j$
Algorithm loop7(n) for $i = 1$ to n $j = 1$ while $j \leq i$ $j = j + 1$	Algorithm loop8(n) for $i = 1$ to n $j = i$ while $j > 0$ $j = j - 1$	Algorithm loop9(n) $i = 0$ $j = 0$ while $i \leq n$ if $i < j$ then $i = i + 1$ else $j = j + 1$ $i = 0$

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

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

loop1	247.1
loop2	247.2
loop3	247.3
loop4	247.4
loop5	247.5
loop6	247.6
loop7	247.7
loop8	247.8
loop9	247.9

Opgave 248

check

Algorithm loop1(n)

```

 $i = 1$ 
while  $i \leq n$ 
     $j = 1$ 
    while  $j \leq i$ 
         $j = j + 1$ 
     $i = i + 1$ 

```

Algorithm loop2(n)

```

 $i = 1$ 
while  $i \leq n$ 
     $j = 1$ 
    while  $j \leq n$ 
         $j = j + 1$ 
     $i = i + 1$ 

```

Algorithm loop3(n)

```

 $i = n$ 
 $j = 0$ 
while  $i > 0$ 
    if  $j < i$ 
         $j = j + 1$ 
    else
         $j = 0$ 
         $i = i - 1$ 

```

Algorithm loop4(n)

```

 $s = 0$ 
for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $n$ 
        if  $i = j$  then
            for  $k = 1$  to  $n$ 
                 $s = s + 1$ 

```

Algorithm loop5(n)

```

 $s = 0$ 
 $i = n$ 
while  $i > 0$ 
    for  $j = 1$  to  $i$ 
         $s = s + 1$ 
     $i = i - 1$ 

```

Algorithm loop6(n)

```

 $s = 1$ 
for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $n$ 
         $s = s + 1$ 

```

Algorithm loop7(n)

```

 $s = 1$ 
for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $n$ 
         $s = s + 1$ 
    for  $k = 1$  to  $n$ 
         $s = s + 1$ 

```

Algorithm loop8(n)

```

 $s = 1$ 
for  $i = 1$  to  $n$ 
    for  $j = i$  to  $n$ 
         $s = s + 1$ 

```

Algorithm loop9(n)

```

 $s = 1$ 
for  $i = n$  to  $1$  step  $-1$ 
    for  $j = n$  to  $1$  step  $-1$ 
         $s = s + 1$ 

```

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

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

loop1	248.1
loop2	248.2
loop3	248.3
loop4	248.4
loop5	248.5
loop6	248.6
loop7	248.7
loop8	248.8
loop9	248.9

Opgave 249

check

```

Algorithm loop1( $n$ )
  for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $i$ 
       $k = 1$ 
      while  $k \leq i + j$ 
         $k = 2 * k$ 

Algorithm loop2( $n$ )
   $s = 0$ 
  for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $i * i$ 
       $s = s + 1$ 

Algorithm loop3( $n$ )
   $s = 0$ 
  for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $n$ 
      for  $k = 1$  to  $n$ 
         $s = s + 1$ 

Algorithm loop4( $n$ )
   $s = 0$ 
  for  $i = 1$  to  $n$ 
    for  $j = i$  to  $n$ 
      for  $k = i$  to  $j$ 
         $s = s + 1$ 

Algorithm loop5( $n$ )
   $s = 0$ 
   $j = 0$ 
  for  $i = 1$  to  $n$ 
     $j = j + i$ 
    for  $k = 1$  to  $j$ 
       $s = s + 1$ 

Algorithm loop6( $n$ )
   $s = 1$ 
  for  $i = 1$  to  $n$ 
    for  $j = 1$  to  $i$ 
      for  $k = j$  to  $i$ 
         $s = s + 1$ 

Algorithm loop7( $n$ )
   $s = 1$ 
  for  $i = 1$  to  $n * n$ 
    for  $j = 1$  to  $n$ 
       $s = s + 1$ 

```

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

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

loop1	249.1
loop2	249.2
loop3	249.3
loop4	249.4
loop5	249.5
loop6	249.6
loop7	249.7

Amortiseret analyse

Opgave 250 (4 %)

check

Antag en liste L bruges til at opbevare en mængde af forskellige tal. Følgende to operationer ønskes understøttet: $\text{ADD}(x)$ 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. $\text{ADD}(x)$ 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 det er en potentialefunktion, hvormed man kan argumentere for at begge operationer tager amortiseret $O(1)$ tid.

	Ja	Nej
$ L $		250.1
$\log L $		250.2
$2 L $		250.3
$ L /2$		250.4
$ L ^2$		250.5
$ L \cdot \log L $		250.6

Opgave 251 (4 %)

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 det er en potentialefunktion, hvormed man kan argumentere for at stakoperationerne PUSH og POP på de to stakke tager amortiseret $O(1)$ tid.

	Ja	Nej
$s+t$		251.1
$t-s$		251.2
$s+n-t$		251.3
$\max\{0, 2(s+t)-n\}$		251.4
$n-s-t$		251.5
$\max\{0, 2s+3t-n\}$		251.6

Opgave 252 (4 %)

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 det er en potentialefunktion, hvormed man kan argumentere for at slettelser tager amortiseret $O(\log n)$ tid.

	Ja	Nej
N		252.1
n		252.2
$N - n$		252.3
$(N - n) \cdot \log n$		252.4
$\log n$		252.5
$(N - n) \cdot \log N$		252.6

Opgave 253 (4 %)

check

Betragt en liste $L = (x_1, \dots, x_N)$ af N heltal, hvorpå vi kan udføre følgende to operationer: 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 det er en potentialefunktion, hvormed man kan argumentere for at begge operationer tager amortiseret $O(1)$ tid.

	Ja	Nej
N		253.1
$\log N$		253.2
$N + \log N$		253.3
$N/2$		253.4
$N \cdot \log N$		253.5
$\sqrt{N}$		253.6

Opgave 254 (4 %)

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 det 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
N		254.1
P		254.2
$2N + P$		254.3
$N + P$		254.4
$N + 2P$		254.5
$P + N/2$		254.6

Opgave 255 (4 %)

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 det er en potentialefunktion, hvormed man kan argumentere for at `INSERTCUT` tager amortiseret $O(\log n)$.

	Ja	Nej
$\log n$		255.1
$n \cdot \log n$		255.2
n		255.3
$\sum_{i=1}^n \log i$		255.4
$k \cdot \log n$		255.5

Opgave 256 (4 %)

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 det er en potentialefunktion, hvormed man kan argumentere for at det totale antal genindsættelser i hashtabellerne er amortiseret $O(1)$ per indsættelse og slettelse i mængden.

	Ja	Nej
$ 2n - N $		256.1
$\min(n, N - n)$		256.2
$N - n$		256.3
$4 N/2 - n $		256.4
N		256.5
$N/2 - \min(n, N - n)$		256.6

Opgave 257 (4 %)

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 det 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
$n \cdot \log n$		257.1
$\sum_{i=1}^n \log i$		257.2
n		257.3
$\log n$		257.4
$(\log n)^2$		257.5

Opgave 258 (4 %)

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 det er en potentialefunktion, hvormed man kan argumentere for at INSERT og HEAP-EXTRACT-MAX kræver amortiseret $O(\log n)$ tid.

	Ja	Nej
n		258.1
$ 2n - N $		258.2
$\max(0, 2n - N)$		258.3
$N - 2n$		258.4
$\log n$		258.5
$\log(N/n)$		258.6

Opgave 259 (4 %)

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.

	$O(1)$	$O(\log n)$	$O(\sqrt{n})$	$O(n)$
$N = n + 1$				259.1
$N = n + \lceil \sqrt{n} \rceil$				259.2
$N = \lceil \frac{3}{2}n \rceil$				259.3
$N = 2n$				259.4
$N = 3n$				259.5

Opgave 260 (4 %)

check

En binær max-heap understøtter MAX-HEAP-INSERT og HEAP-EXTRACT-MAX i worst-case tid $O(\log n)$, hvor n er antal elementer i heapen. Vi vil nu understøtte operationen DELETE, der givet en pointer til et element i heapen, sletter elementet fra heapen. Vi implementerer DELETE ved blot at markere elementet som slettet i worst-case $O(1)$ tid. Når vi udfører HEAP-EXTRACT-MAX gentager vi denne indtil det første ikke-markerede element bliver returneret. Dvs. hvis HEAP-EXTRACT-MAX sletter D markerede elementer bliver worst-case tiden $O((D+1) \log N)$, hvor N er antallet af markerede og ikke-markerede elementer i heapen.

Angiv for hver af nedenstående funktioner om det er en potentialefunktion, hvormed man kan argumentere for at operationerne MAX-HEAP-INSERT, DELETE, og HEAP-EXTRACT-MAX tager amortiseret $O(\log N)$ tid, hvor M betegner antallet af markerede elementer i heapen.

	Ja	Nej
N		260.1
M		260.2
$N \cdot \log N$		260.3
$M \cdot \log N$		260.4
$N \cdot \log M$		260.5
$M \cdot \log M$		260.6

Opgave 261 (4 %)

check

En binær max-heap understøtter INSERT og HEAP-EXTRACT-MAX på en heap med n elementer i worst-case $O(\log n)$ tid. Bemærk den samme værdi kan være indsat flere gange i en max-heap. Vi ønsker nu at ændre HEAP-EXTRACT-MAX, således at den fjerner alle forekomster af maximum værdien fra heapen, d.v.s. den oprindelige HEAP-EXTRACT-MAX operation gentages indtil roden indeholder en mindre værdi eller at heapen er tom. Hvis maximum forekommer m gange i heapen, så vil HEAP-EXTRACT-MAX operationen tage worst-case $O(m \log n)$ tid.

Angiv for hver af nedenstående funktioner om det er en potentialefunktion, hvormed man kan argumentere for at operationerne INSERT og HEAP-EXTRACT-MAX tager amortiseret $O(\log n)$ tid. Antal elementer i heapen betegnes n og antallet af forskellige elementer i heapen N , hvor $N \leq n$.

	Ja	Nej
n		261.1
N		261.2
$n - N$		261.3
$n \cdot \log n$		261.4
$N \cdot \log n$		261.5
$(n - N) \cdot \log n$		261.6

Invarianter

Opgave 262 (4 %)

check

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

Algorithm Multiplication(n, m)
 Input : $n \geq 0 \wedge m \geq 0$
 Output : $r = n_0 \cdot m_0$
 Method : $r \leftarrow 0$
 $\{I\}$ **while** $n > 0$ **do**
 if n is odd **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 det er en invariant I for algoritmen Multiplication, hvor n_0 og m_0 angiver værdierne for henholdsvis n og m i starten.

Ja Nej

$0 \leq n \leq n_0$	262.1
$0 \leq m \leq m_0$	262.2
$r = m \cdot n$	262.3
$n_0 \cdot m_0 = r + n \cdot m$	262.4
$n_0 \cdot m_0 + r = n \cdot m$	262.5

Opgave 263 (4 %)

check

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

Algorithm Power(x, y)
 Input : $x \geq 1 \wedge y \geq 1$
 Output : $r = x^y$
 Method : $r \leftarrow 1$
 $\{I\}$ **while** $y \geq 1$ **do**
 if y is odd **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 det er en invariant I for algoritmen Power, hvor x_0 og y_0 angiver start værdierne for x og y .

Ja Nej

$r = x^y$	263.1
$r = x_0^{y_0}$	263.2
$r = x_0^{y_0 - y}$	263.3
$x_0^{y_0} = r \cdot x^y$	263.4
$x^y = r \cdot x_0^{y_0}$	263.5

Opgave 264 (4 %)

check

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

Algorithm Square(n)
 Input : $n \geq 1$
 Output : $r = n^2$
 Method : $r \leftarrow 0$
 $i \leftarrow 0$
 $\{I\}$ **while** $i < n$ **do**
 $i \leftarrow i + 1$
 $r \leftarrow r + i$
 $r \leftarrow 2 * r - n$

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

Ja Nej

$i \geq 0 \wedge r = i^2$	264.1
$0 \leq i \leq n$	264.2
$i \geq 0 \wedge r = i(i + 1)/2$	264.3
$r \geq i \geq 0$	264.4
$0 \leq r \leq n$	264.5

Opgave 265 (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. $\text{count}(x, A) = |\{i \mid 1 \leq i \leq n \wedge A[i] = x\}|$.

```

Algorithm Count( $x, A$ )
Input   :  $x \wedge A[1..n] \wedge n \geq 1$ 
Output  :  $r = \text{count}(x, A)$ 
Method :  $i \leftarrow 0$ 
          $r \leftarrow 0$ 
         {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 det er en invariant I for ovenstående algoritme Count.

Ja Nej

- | | |
|------------------------------------|-------|
| $i \leq n$ | 265.1 |
| $r = \text{count}(x, A[1..i])$ | 265.2 |
| $r = \text{count}(x, A[1..i + 1])$ | 265.3 |
| $i < n$ | 265.4 |
| $r = 0$ | 265.5 |

Opgave 266 (4 %)

check

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

Algorithm FindDiff($A[1..n], x$)
Input : $x > 0 \wedge A[1..n] \wedge n \geq 2 \wedge A[1] < \dots < A[n]$
Output : $1 \leq i < j \leq n$, where $x = A[j] - A[i]$;
 or $j = n + 1$ if no such pair exists
Method : $i \leftarrow 1$;
 $j \leftarrow 1$;
 { 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 det er en invariant I for algoritmen FindDiff.

	Ja	Nej
$i \leq j$		266.1
$1 \leq i \leq j \leq n$		266.2
$x \neq A[j] - A[i]$		266.3
$\forall 1 \leq i' < j' \leq n : (i' \geq i \vee j' \geq j) \Rightarrow x \neq A[j'] - A[i']$		266.4
$\forall 1 \leq i' < j' \leq n : (i' < i \vee j' < j) \Rightarrow x \neq A[j'] - A[i']$		266.5

Opgave 267 (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 $\|x\|$.

Algorithm Bits(x)
 Input : $x \geq 0$
 Output : $r = \|x\|$
 Method : $r \leftarrow 0$;
 $\{I\}$ **while** $x > 0$ **do**
 if x is odd **then**
 $x \leftarrow x - 1$;
 $r \leftarrow r + 1$
 $x \leftarrow x/2$

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

Ja Nej

$r = \ x\ $	267.1
$r + \ x\ = \ x_0\ $	267.2
$r + \ x_0\ = \ x\ $	267.3
$r + \ x\ \leq x_0$	267.4
$\ x\ + 2^r = x_0$	267.5

Opgave 268 (4 %)

check

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

Algorithm Log2(n)
 Input : $n \geq 2$
 Output : $r = \text{intlog}(n) = \lfloor \log_2 n \rfloor$
 Method : $i \leftarrow 1$;
 $r \leftarrow 1$;
 $p \leftarrow 2$;
 $\{I\}$ **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 det er en invariant I for algoritmen Log2.

Ja Nej

$1 \leq r < p$	268.1
$2p \leq n$	268.2
$p = 2^r$	268.3
$p = 2r$	268.4
$p = 2^{\text{intlog}(p)}$	268.5

Opgave 269 (4 %)

check

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

Algorithm Missing($A[1..n-1]$)

Input : $A[1..n-1] \wedge n \geq 2 \wedge A[i] \in \{1, 2, \dots, n\} \wedge A[i] \neq A[j]$ for $1 \leq i < j < n$

Output : $r \in \{1, 2, 3, \dots, n\} \setminus A$

Method : $i \leftarrow 1$;

$x \leftarrow 1$;

$y \leftarrow A[1]$;

{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 det er en invariant I for algoritmen Missing.

Ja Nej

$1 \leq i \leq n-1$ 269.1

$y = x - i + A[i]$ 269.2

$x = i(i+1)/2$ 269.3

$y = \sum_{j=1}^i A[j]$ 269.4

$i = i + 1$ 269.5

Opgave 270 (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 $\text{dom}(A)$ angive antal dominerende positioner i i A , hvor $1 \leq i \leq n$. Nedenstående algoritme til venstre beregner $\text{dom}(A)$.

```

Algorithm Domination( $A$ )
Input   : Array  $A[1..n] \wedge n \geq 1$ 
Output  :  $r = \text{dom}(A)$ 
Method :  $i \leftarrow 1$ ;
          $x \leftarrow A[1]$ ;
          $r \leftarrow 1$ ;
         { $I$ } 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 det er en invariant I for algoritmen.

Ja Nej

$1 \leq i < n$	270.1
$x = A[i]$	270.2
$x \geq A[i]$	270.3
$r = \text{dom}(A[1..n])$	270.4
$r = \text{dom}(A[1..i])$	270.5

Opgave 271 (4 %)

check

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

Algorithm Squares($A[1..n]$)
 Input : Array $A[1..n] \wedge 0 < A[1] < A[2] < \dots < A[n]$
 Output : $r = \text{squares}(A)$
 Method : $i \leftarrow 1$;
 $j \leftarrow 1$;
 $r \leftarrow 0$;
 { I } **while** $j \leq n$ **do**
 if $A[i]^2 < A[j]$ **then**
 $i \leftarrow i + 1$
 else if $A[i]^2 = A[j]$ **then**
 $i \leftarrow i + 1$; $j \leftarrow j + 1$; $r \leftarrow r + 1$
 else if $A[i]^2 > A[j]$ **then**
 $j \leftarrow j + 1$

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

Ja Nej

$1 \leq i \leq j \leq n$	271.1
$i \leq j$	271.2
$A[i-1]^2 < A[j] \wedge r = \text{squares}(A[1..j])$	271.3
$A[i-1]^2 < A[j] \wedge r = \text{squares}(A[1..j-1])$	271.4
$r = j - 1$	271.5

Opgave 272 (4 %)

check

Givet et positivt heltal n , så beregner nedenstående algoritme n^3 .

Algorithm Power3(n)
 Input : $n \geq 1$
 Output : $r = n^3$
 Method : $i \leftarrow 1$
 $s \leftarrow 1$
 $r \leftarrow 1$
 { I } **while** $i < n$ **do**
 $i \leftarrow i + 1$
 $s \leftarrow s + 2i - 1$
 $r \leftarrow r + 3s - 3i + 1$

For hvert af følgende udsagn, angiv om det er en invariant I for algoritmen Power3.

Ja Nej

$1 \leq i < n$	272.1
$1 \leq i \leq s \leq r$	272.2
$r = i^3$	272.3
$r = n^3$	272.4
$s = s + 2i - 1$	272.5

Opgave 273 (4 %)

check

Givet et ikke-negativt heltal x og et positive heltal y , så beregner nedenstående algoritme $\lfloor x/y \rfloor$.

Algorithm Division(x, y)
 Input : $x \geq 0 \wedge y \geq 1$
 Output : $r = \lfloor x/y \rfloor$
 Method : $r \leftarrow 0$
 { I } **while** $x \geq y$ **do**
 $x \leftarrow x - y$
 $r \leftarrow r + 1$

For hvert af følgende udsagn, angiv om det er en invariant I for algoritmen Division, hvor x_0 og y_0 angiver værdierne for henholdsvis x og y i starten.

Ja Nej

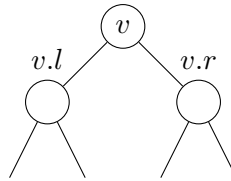
$r = \lfloor x/y \rfloor$	273.1
$r = \lfloor x_0/y_0 \rfloor$	273.2
$r = \lfloor (x_0 - x)/y \rfloor$	273.3
$x + ry = x_0$	273.4
$r(x - x_0) = y$	273.5

Udvidede søgetræer

Opgave 274 (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 *præfix* 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).



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

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

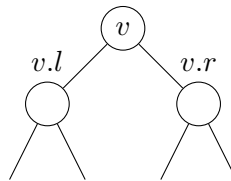
$$v.prefix = \max(v.l.prefix, v.W + v.r.prefix)$$

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

Opgave 275 (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 maksimale 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).



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

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

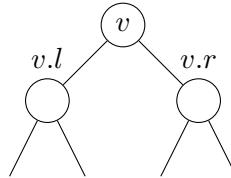
$$v.max = \max(v.r.max, v.l.max, v.high)$$

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

Opgave 276 (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 henholdsvis 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).



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

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

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

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

$$v.avg = (v.x + v.l.avg + v.r.avg)/3$$

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

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

$$v.avg = v.x + (v.l.avg + v.r.avg)/2$$

Opgave 277 (4 %)

check

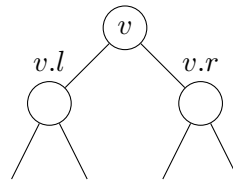
Betragt en liste af n punkter $(x_1, y_1), \dots, (x_n, y_n)$, hvor x_i og y_i er reelle tal og $x_1 < x_2 < \dots < x_n$. Vi ønsker at finde en sammenhængende række af punkter

$$(x_i, y_i), (x_{i+1}, y_{i+1}), \dots, (x_{j-1}, y_{j-1}), (x_j, y_j),$$

hvor $i \leq j$, således at $Y_{i,j} = \sum_{k=i}^j y_k = y_i + y_{i+1} + \dots + y_{j-1} + y_j$ 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 indeholder punkterne $(x_k, y_k), \dots, (x_\ell, y_\ell)$ fire værdier $v.sum = Y_{k,\ell}$, $v.maxysum = \max_{k \leq i \leq \ell} Y_{i,j}$, $v.pre = \max\{0, \max_{k \leq j \leq \ell} Y_{k,j}\}$, 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).



$$v.sum = v.l.sum + v.r.sum$$

$$v.sum = v.l.sum + v.y + v.r.sum$$

$$v.sum = v.l.suf + v.r.pre$$

$$v.sum = v.l.suf + v.y + v.r.pre$$

$$v.maxysum = \max\{v.l.maxysum, v.r.maxysum\}$$

$$v.maxysum = \max\{v.l.maxysum, v.y, v.r.maxysum\}$$

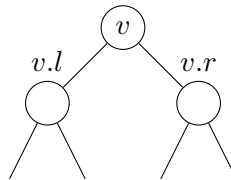
$$v.maxysum = \max\{v.l.maxysum, v.l.suf + v.y + v.r.pre, v.r.maxysum\}$$

$$v.maxysum = \max\{v.l.maxysum, v.l.suf + v.r.pre, v.r.maxysum\}$$

Opgave 278 (4 %)**check**

Betragt en liste af n punkter $(x_1, y_1), \dots, (x_n, y_n)$, hvor x_i og y_i er reelle tal og $x_1 < x_2 < \dots < x_n$. 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).



$$v.miny = v.l.miny$$

$$v.miny = \min\{v.l.miny, v.r.miny\}$$

$$v.miny = \min\{v.l.miny, v.y, v.r.miny\}$$

$$v.DF = v.l.DF \wedge v.r.DF$$

$$v.DF = v.l.DF \wedge v.r.DF \wedge v.y < v.l.miny \wedge v.y > v.r.maxy$$

$$v.DF = v.l.DF \wedge v.r.DF \wedge v.y > v.l.miny \wedge v.y < v.r.maxy$$

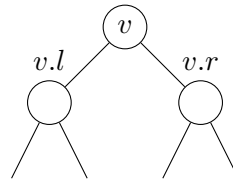
$$v.DF = v.l.DF \wedge v.r.DF \wedge v.y < v.l.miny$$

Opgave 279 (4 %)

check

Betragt en liste af n par $(x_1, c_1), \dots, (x_n, c_n)$, hvor $x_1 < x_2 < \dots < x_n$ 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).



$$v.mono = v.l.mono \wedge v.r.mono$$

$$v.mono = v.l.mono \wedge v.r.mono \wedge v.l.c = v.r.c \wedge v.c = v.r.c$$

$$v.mono = v.l.mono \wedge v.r.mono \wedge v.l.c = v.r.c$$

$$v.mono = v.l.mono \wedge v.r.mono \wedge v.mono$$

$$v.missing = v.l.missing \cup v.r.missing$$

$$v.missing = v.l.missing \cap v.r.missing$$

$$v.missing = \{v.c\} \cup v.l.missing \cup v.r.missing$$

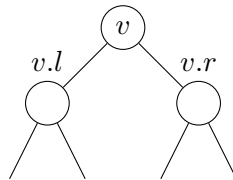
$$v.missing = (v.l.missing \cap v.r.missing) \setminus \{v.c\}$$

Opgave 280 (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 $T = \text{"a)b(cd(x)dc(a)}$ gemmes i træet $\langle v.p, v.c \rangle$ parrene $\langle 2, \rangle \rangle$, $\langle 4, (\rangle$, $\langle 7, (\rangle$, $\langle 9, \rangle \rangle$ og $\langle 12, (\rangle$.

Vi ønsker at vedligeholde information om parenteserne er balancerede. I ovenstående eksempel er parenteserne “) (() (” ikke balancerede, da kun de markerede parenteser går ud mod hinanden. De restende parenteser “) ((” 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).



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

$$v.R = v.l.R + v.r.R + 1 - v.l.L$$

$$v.R = v.l.R + \max\{0, v.r.R + 1 - v.l.L\}$$

$$v.R = v.l.R + 1 + \max\{0, v.r.R - v.l.L\}$$

Opgave 281 (4 %)

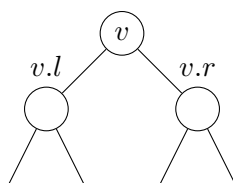
check

For n tal $x_1, \dots, x_n$ 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 koefficienterne 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).



$$v.a = v.l.a + v.r.a + v.x$$

$$v.a = v.l.a + v.r.a - 2 * v.x$$

$$v.a = v.l.a * v.r.a * (v.x)^2$$

$$v.a = v.l.a + v.r.a + (v.x)^2$$

$$v.b = v.l.b + v.r.b + v.x$$

$$v.b = v.l.b + v.r.b + (v.x)^2$$

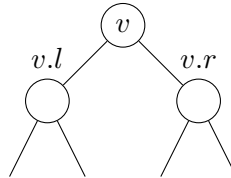
$$v.b = v.l.b + v.r.b - 2 * v.x$$

$$v.b = v.l.b + v.r.b + 2 * v.x$$

Opgave 282 (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 tre værdier $v.min$, $v.max$, og $v.closest$. Værdierne $v.min$ og $v.max$ er henholdsvis det mindst og største tal i v 's undertræ, og $v.closest$ den mindste difference mellem to tal i v 's undertræ. Hvis v 's undertræ kun indeholder et tal er $v.closest = +\infty$. Angiv hvorledes $v.closest$ og kan beregnes når min , max og $closest$ værdierne er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).



$$v.closest = \min(v.l.closest, v.r.x - v.l.x, v.r.closest)$$

$$v.closest = \min(v.l.closest, v.x - v.l.min, v.r.max - v.x, v.r.closest)$$

$$v.closest = \min(v.l.closest, v.x - v.l.max, v.r.min - v.x, v.r.closest)$$

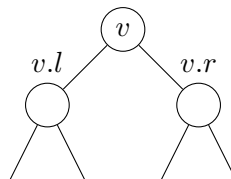
$$v.closest = \min(v.l.closest, v.r.min - v.l.max, v.r.closest)$$

$$v.closest = v.r.closest - v.l.closest$$

Opgave 283 (4 %)

check

For en sorteret liste af tal $x_1 \leq x_2 \leq \dots \leq x_n$ definerer vi *sum of square gaps* (ssg) som $\sum_{i=2..n} (x_i - x_{i-1})^2$. Betrakt et rød-sort træ hvor hver knude v gemmer et heltal $v.x$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Desuden gemmer v værdierne $v.min$, $v.max$ og $v.ssg$, som er hhv. det mindste, største og *sum of square gaps* af elementerne i undertræet rodet i v . Angiv hvorledes $v.ssg$ kan beregnes når $v.min$, $v.max$ og $v.ssg$ er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).



$$v.ssg = v.l.ssg + v.r.ssg$$

$$v.ssg = v.l.ssg + (v.r.min - v.l.max)^2 + v.r.ssg$$

$$v.ssg = v.l.ssg + (v.x - v.l.max)^2 + (v.x - v.r.min)^2 + v.r.ssg$$

$$v.ssg = v.l.ssg + (v.r.x - v.l.x)^2 + v.r.ssg$$

$$v.ssg = v.l.ssg + (v.x - v.l.x)^2 + (v.x - v.r.x)^2 + v.r.ssg$$

Diverse spørgsmål

Opgave 284 (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.

$\Theta(\log n)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$ $\Theta(2^n)$

Worst-case tid for INSERTION-SORT	284.1
Worst-case tid for MERGE-SORT	284.2
Worst-case tid for HEAPSORT	284.3
Worst-case tid for QUICKSORT	284.4
Best-case tid for INSERTION-SORT	284.5
Best-case tid for MERGE-SORT	284.6
Best-case tid for HEAPSORT	284.7
Best-case tid for QUICKSORT	284.8
Forventet tid for INSERTION-SORT	284.9
Forventet tid for MERGE-SORT	284.10
Forventet tid for HEAPSORT	284.11
Forventet tid for QUICKSORT	284.12

Opgave 285 (4 %)

check

Angiv worst-case tiden for HEAPSORT på et array med n *identiske* elementer.

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

Opgave 286 (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.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$

Opgave 287 (4 %)

check

Givet et sorteret array $A[1..n]$ ($A[1] < A[2] < \dots < A[n]$) og et element x , så ønsker vi at finde indekset ℓ således at $A[\ell] \leq x < A[\ell + 1]$. Det antages at $A[1] \leq x < A[n]$. 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 $\ell + 1 < h$	while $\ell < h$	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 $A[m] > x$	if $A[m] \leq x$	if $A[m] \leq x$
$\ell = m$	$\ell = m$	$\ell = m$	$\ell = m$
else	else	else	else
$h = m$	$h = m$	$h = m$	$h = m$

Opgave 288 (4 %)

check

Strassens algoritme til multiplikation af kvadratiske $n \times n$ matricer er en del-og-kombiner algoritme. Angiv hvilken rekursionsligning der beskriver udførselstiden af Strassen's algoritme

$$T(n) \leq 7 \cdot T(n/2) + c \cdot n$$

$$T(n) \leq 7 \cdot T(n/4) + c \cdot n^2$$

$$T(n) \leq 7 \cdot T(n/2) + c \cdot n^2$$

$$T(n) \leq 7 \cdot T(n/2) + c \cdot n^3$$

Opgave 289 (4 %)

check

Angiv worst-case udførselstiden for hver af følgende algoritmer når input er et array af størrelse n .

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^3)$	
BINARY-SEARCH								289.1
INSERTION-SORT								289.2
MERGE-SORT								289.3
HEAPSORT								289.4
QUICKSORT								289.5
PARTITION								289.6

Opgave 290 (4 %)

check

Givet et array af størrelse n indeholdende tallene $1, 2, \dots, n$ i voksende rækkefølge, hvad er worst-case tiden for følgende sorteringsalgoritmer, når de anvendes på arrayet?

	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	
INSERTION-SORT				290.1
HEAPSORT				290.2
MERGE-SORT				290.3
QUICKSORT				290.4

Opgave 291 (4 %)

check

Selektionsalgoritmen til at finde det *ite* 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?

$$\Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

Opgave 292 (4 %)

check

For en binær max-heap af størrelse n , angiv best-case og worst-case udførselstid for følgende operationer.

$$\Theta(1) \quad \Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n)$$

INSERT, worst-case	292.1
INSERT, best-case	292.2
HEAP-EXTRACT-MAX, worst-case	292.3
HEAP-EXTRACT-MAX, best-case	292.4
BUILD-MAX-HEAP, worst-case	292.5
BUILD-MAX-HEAP, best-case	292.6

Opgave 293 (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.

$$\Theta(1) \quad \Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

Worst-case antal sammenligninger	293.1
Best-case antal sammenligninger	293.2
Forventet antal sammenligninger	293.3

Opgave 294 (4 %)**check**

Hvilke af følgende udsagn er sande for alle binære søgetræer med n elementer.

Ja Nej

Elementet i et blad er altid $\leq$ elementet i roden	294.1
Elementer med samme dybde i træet er sorteret fra venstre-mod-højre	294.2
Længste rod-til-blad sti indeholder $O(\log n)$ knuder	294.3
Det højre barn til en knude med rang r har rang $r + 1$	294.4
Langs stien fra roden til det største element, er elementerne voksende	294.5

Opgave 295 (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 ?

$O(\log n)$ $O(n)$ $O(n \log n)$ $O(n^2)$ $O(n\sqrt{n})$ $O(\sqrt{n})$ $O((\log n)^2)$

Opgave 296 (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

$Q.tail - Q.head$	296.1
$ Q.tail - Q.head $	296.2
$ Q.tail - Q.head \bmod Q.length$	296.3
$(Q.tail - Q.head + Q.length - 1) \bmod Q.length$	296.4
$(Q.tail - Q.head + Q.length) \bmod Q.length$	296.5

Opgave 297 (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

$a \leq b$	297.1
$b \leq c$	297.2
$a \leq c$	297.3

Opgave 298 (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?

$$\Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

Opgave 299 (4 %)

check

Angiv for hver af nedenstående summer deres værdi i Θ -notationen. Det antages at n er en potens af to.

$$\Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

$$\sum_{i=1}^n i = 1 + 2 + 3 + \cdots + n \quad 299.1$$

$$\sum_{i=0}^{\log n} 2^i = 1 + 2 + 4 + 8 + \cdots + n \quad 299.2$$

$$\sum_{i=1}^{\log n} i \frac{n}{2^i} = 1 \frac{n}{2^1} + 2 \frac{n}{2^2} + 3 \frac{n}{2^3} + \cdots + \log n \frac{n}{2^{\log n}} \quad 299.3$$

$$\sum_{i=1}^n \log i = \log 1 + \log 2 + \log 3 + \cdots + \log n \quad 299.4$$

$$\sum_{i=0}^{\log n} \frac{n}{2^i} = n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + \frac{n}{n} \quad 299.5$$

Opgave 300 (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 .

$$\Theta(d) \quad \Theta(\log_2 n) \quad \Theta(\log_d n) \quad \Theta(d \log_2 n) \quad \Theta(d \log_d n)$$

$$\text{INSERT} \quad 300.1$$

$$\text{EXTRACT-MAX} \quad 300.2$$

Opgave 301 (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?

$$1 \quad n \quad k \quad k-1 \quad 2^{k-1} \quad 2^k - 1$$

Opgave 302 (4 %)

check

Hvor mange gange kaldes RANDOMIZED-PARTITION i RANDOMIZED-SELECT på et array af størrelse n ?

$$\Theta(1) \quad \Theta(\log n) \quad \Theta(\sqrt{n}) \quad \Theta(n) \quad \Theta(n \log n)$$

Forventede antal kald	302.1
Færrest antal kald (best case)	302.2
Flest antal kald (worst case)	302.3

Opgave 303 (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?

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

Opgave 304 (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 $A[1..n]$?

$$\Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

Opgave 305 (4 %)

check

Antag MERGE-SORT udføres på et input af størrelse n og indeholdende to elementer x og y . Hvad er worst-case antal sammenligninger af x med y under udførelsen af MERGE-SORT?

$$\Theta(1) \quad \Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2)$$

Opgave 306 (4 %)

check

Antag vi har en orienteret graf med n knuder og positivt vægtede kanter, hvor vi løbende tilføjer yderligere kanter. Vi ønsker at vedligeholde en afstands-tabel over de korteste afstande mellem alle par af knuder.

Hvad er den bedste worst-case tid man kan opnå for at opdatere afstands-tabellen, når man tilføjer en ny kant med positiv vægt til grafen?

$$\Theta(1) \quad \Theta(\sqrt{n}) \quad \Theta(n) \quad \Theta(n \cdot \log n) \quad \Theta(n \cdot \sqrt{n}) \quad \Theta(n^2) \quad \Theta(n^3)$$