

# EKSAMEN

## Algoritmer og Datastrukturer

(indsæt dato og tid)

Institut for Datalogi, Naturvidenskabelige Fakultet, Aarhus Universitet

Antal sider i opgavesættet (incl. forsiden): 122

Tilladte medbragte hjælpemidler: **Ingen**

**Studienummer :**

**Navn :**

compute score (max 920 point)

## Vejledning og pointgivning

Dette eksamenssæt består af en mængde multiple-choice-opgaver.

Opgaverne besvares på opgaveformuleringen **som afleveres**.

For hver opgave er angivet opgavens andel af det samlede eksamenssæt.

Hvert delspørgsmål har præcist et rigtigt svar.

For hvert delspørgsmål må du vælge **max ét svar** ved at afkrydse den tilsvarende rubrik.

Et delspørgsmål bedømmes som følgende:

- Hvis du sætter kryds ved det rigtige svar, får du 1 point.
- Hvis du ikke sætter nogen krydser, får du 0 point.
- Hvis du sætter kryds ved et forkert svar, får du  $-\frac{1}{k-1}$  point, hvor  $k$  er antal svarmuligheder.

For en opgave med vægt  $v\%$  og med  $n$  delspørgsmål, hvor du opnår samlet  $s$  point, beregnes din besvarelse af opgaven som:

$$\frac{s}{n} \cdot v \%$$

Bemærk at det er muligt at få negative point for en opgave.

**Opgave 1 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$4 \text{ er } O(1)$$

$$3 \log n^2 \text{ er } O((\log n)^2)$$

$$5n^{1/3} \text{ er } O(2^n)$$

$$3^3/7 \text{ er } O(n^2)$$

$$2^{2 \log n} \text{ er } O(\log(n!))$$

$$n \cdot \log n \text{ er } O(\sqrt{n} \cdot \log n)$$

$$n/5 + 4 \log n \text{ er } O(n^{1/3})$$

$$n^{0.1} \text{ er } O(\sqrt{n})$$

$$4^{\log n} \text{ er } O(n)$$

$$n^3 \text{ er } \Omega(n \cdot \log n)$$

$$1 \text{ er } \Theta(n^2)$$

$$4 \log n^2 \text{ er } \Omega(n^2)$$

**Opgave 2 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$n \text{ er } O(5)$$

$$\log n^2 + n \cdot \log n \text{ er } O(8^{\log n})$$

$$2^{2 \log n} \text{ er } O(n^2)$$

$$(\log n)^3 + n^{0.01} \text{ er } O(\log n)$$

$$(\log n)/6 \text{ er } O(8^{\log n})$$

$$n \cdot \log n \text{ er } O(n^3)$$

$$4(\log n)^6 \text{ er } O(1)$$

$$(\log n)^7 \text{ er } O(n \cdot \log n)$$

$$7\sqrt{n} \cdot \log n \text{ er } O(\log n)$$

$$3 \log n \text{ er } \Theta(\log n)$$

$$\sqrt{n} \text{ er } \Omega(4^{\log n})$$

$$1 \text{ er } \Omega(\sqrt{n})$$

**Opgave 3 (Asymptotisk notation, 6 %)**

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

Ja Nej

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

**Opgave 4 (Asymptotisk notation, 6 %)**

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

Ja Nej

- $\sqrt{n} \cdot \log n + n^n$  er  $O(2^{\log n})$
- $n^3/2$  er  $O(n^{3/2})$
- $4^{\log n}$  er  $O(\sqrt{n})$
- $n \cdot \log n$  er  $O(n^{3/2})$
- $\sqrt{n}$  er  $O(n^{2/3})$
- $\sqrt{n}$  er  $O(n!)$
- $1$  er  $O((\log n)^2)$
- $n^{3/2}$  er  $O(\sqrt{n})$
- $4\sqrt{n}$  er  $O(8^{\log n})$
- $\sqrt{n}$  er  $\Theta(n^2)$
- $7n^{0.01}$  er  $\Omega((\log n)^6)$
- $2\sqrt{n}$  er  $\Theta(n^{3/2})$

**Opgave 5 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$3n \cdot \log n \text{ er } O(n \cdot \log n)$$

$$3^3/6 \text{ er } O(n^2)$$

$$n \cdot \log n + n \cdot \log n \text{ er } O(3^n)$$

$$2^2 \text{ er } O(n \cdot \log n)$$

$$(\log n)^3 \text{ er } O(\log n)$$

$$n^{0.001} \text{ er } O(n^{0.1})$$

$$3^n \text{ er } O(n^2)$$

$$n^{0.1} + n^{2/3} \text{ er } O(n^2 \log n)$$

$$\log(n!) + \log n^2 \text{ er } O(n^3)$$

$$8^{\log n} + 4\sqrt{n} \cdot \log n \text{ er } \Theta(2^{3 \log n})$$

$$n^{0.1} \text{ er } \Omega(n^3)$$

$$n! \text{ er } \Omega(\sqrt{n})$$

**Opgave 6 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$n^2 \text{ er } O(n^{1/3})$$

$$n^{2/3} \text{ er } O(2^{\log n})$$

$$n \cdot \log n \text{ er } O(n^{2/3} \cdot n^{1/3})$$

$$2 \text{ er } O(2^{\log n})$$

$$n \cdot \log n \text{ er } O(n^2 \log n)$$

$$n^{3/2} \text{ er } O(4)$$

$$8^{\log n} \text{ er } O(\sqrt{n})$$

$$(\log n)^4 \text{ er } O(n^{1/3})$$

$$7(\log n)^3 + \log n^2 \text{ er } O(n\sqrt{n})$$

$$n^3/4 \text{ er } \Theta(2^{3 \log n})$$

$$\sqrt{n} \cdot \log n + 1 \text{ er } \Omega(n)$$

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

**Opgave 7 (Asymptotisk notation, 6 %)**

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

Ja Nej

- $7 \log(n!) \text{ er } O(n!)$
- $(\log n)^2 \text{ er } O(n \cdot \log n)$
- $n \cdot \log n + (\log n)^4/4 \text{ er } O(\sqrt{n})$
- $n^{0.001} \text{ er } O((\log n)^6)$
- $n^{2/3} \cdot n^{1/3} \text{ er } O(\log n)$
- $n\sqrt{n}/7 \text{ er } O(n^{0.001})$
- $\sqrt{n} \cdot \log n \text{ er } O(n \cdot \log n)$
- $n^{0.01}/6 \text{ er } O(n^2)$
- $n^2 \text{ er } O(n)$
- $n \cdot \log n \text{ er } \Omega(4^4)$
- $5n^n \text{ er } \Omega(2^2)$
- $n^2(\log n)/6 \text{ er } \Omega(n^{0.01})$

**Opgave 8 (Asymptotisk notation, 6 %)**

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

Ja Nej

- $n! \text{ er } O(\log n)$
- $1 + 6\sqrt{n} \cdot \log n \text{ er } O(\log n^2)$
- $n! \text{ er } O(\sqrt{n})$
- $2^{\log n} \text{ er } O(n \cdot \log n)$
- $n^2 \text{ er } O(\log n^2)$
- $n \cdot \log n \text{ er } O(n^{0.001})$
- $n^{0.1} \text{ er } O(n^2)$
- $n^n \text{ er } O(n^2)$
- $n \text{ er } O((\log n)^2)$
- $3 \text{ er } \Theta(n^2)$
- $(\log n)^2 \text{ er } \Theta(2^n)$
- $n \cdot \log n \text{ er } \Omega(1)$

**Opgave 9 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$2n^n \text{ er } O((\log n)^7)$$

$$5n^{1/3} \text{ er } O(n^2 \log n)$$

$$\sqrt{n} \text{ er } O((\log n)^2)$$

$$n^{0.001} + \sqrt{n} \text{ er } O(n^2 \log n)$$

$$n^{3/2} \text{ er } O((\log n)^5)$$

$$7 \cdot 2^n + \sqrt{n} \text{ er } O(n \cdot \log n)$$

$$n! \text{ er } O(n^{0.01})$$

$$2^{2 \log n} \text{ er } O(n^{1/3})$$

$$5n^{3/2} \text{ er } O(\log n^2)$$

$$2n^2 \log n \text{ er } \Theta(n^2 \log n)$$

$$\sqrt{n} \text{ er } \Omega(n^n)$$

$$2n^{3/2} + n \cdot \log n \text{ er } \Theta(n^{3/2})$$

**Opgave 10 (Asymptotisk notation, 6 %)**

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

Ja Nej

$$2 \cdot 3^n \text{ er } O(n^{1/3})$$

$$n/7 \text{ er } O(n!)$$

$$n^{0.001} + n^{2/3} \text{ er } O(n)$$

$$n^{2/3} \cdot n^{1/3} \text{ er } O(\log n^2)$$

$$n \cdot \log n + n^3 \text{ er } O(n)$$

$$(\log n)^3 \text{ er } O(n^2)$$

$$\log(n!) \text{ er } O(n^{1/3})$$

$$n^{3/2} \text{ er } O(2^n)$$

$$n^2 \text{ er } O(n \cdot \log n)$$

$$n^{1/3} \text{ er } \Omega(\sqrt{n} \cdot \log n)$$

$$2^{3 \log n} \text{ er } \Omega(\sqrt{n} \cdot \log n)$$

$$\log n \text{ er } \Theta(\log n^2)$$

Opgave 11 (Analyse af løkker, 6%)

<p><b>Algoritme loop1(<math>n</math>)</b>  <math>i = n</math>  <b>while</b> <math>i &gt; 0</math>          <math>i = i - 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <math>i = 3 * i</math></p>
<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <math>j = n</math>          <b>while</b> <math>j &gt; 1</math>              <math>j = j - 1</math>          <math>i = 2 * i</math></p>	<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>j = 1</math>  <math>s = 0</math>  <b>while</b> <math>i \leq n</math>          <b>if</b> <math>i = j</math> <b>then</b>              <b>for</b> <math>k = 1</math> <b>to</b> <math>n</math>                  <math>s = s + 1</math>                  <math>j = 2 * j</math>          <math>i = i + 1</math></p>

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4



## Opgave 12 (Analyse af løkker, 6%)

<b>Algoritme loop1(<math>n</math>)</b>	<b>Algoritme loop2(<math>n</math>)</b>
$i = 1$	$s = 1$
<b>while</b> $i \leq n$	<b>while</b> $s \leq n$
$j = 1$	$s = s + 1$
<b>while</b> $j \leq n$	
$j = j + 1$	
$i = i + 1$	
<b>Algoritme loop3(<math>n</math>)</b>	<b>Algoritme loop4(<math>n</math>)</b>
$i = 1$	$i = 1$
<b>while</b> $i * i \leq n$	$s = 0$
$i = i + i$	<b>while</b> $i \leq n$
	<b>for</b> $j = i$ <b>to</b> $n$
	$s = s + 1$
	$i = i + i$

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

Opgave 13 (Analyse af løkker, 6%)

<p><b>Algoritme loop1(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = i</math>              <b>while</b> <math>j &gt; 0</math>                  <math>j = j - 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>s = 1</math>  <b>for</b> <math>i = n</math> <b>to</b> <math>1</math> <b>step</b> <math>-1</math>              <math>s = s + 1</math></p>
---	---

<p><b>Algoritme loop3(<math>n</math>)</b>  <math>j = n</math>  <math>i = 1</math>  <b>while</b> <math>j \geq 0</math>              <math>j = j - i</math>              <math>i = i + 1</math></p>	<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>s = 0</math>  <b>while</b> <math>s \leq n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq i</math>                  <math>j = j + 1</math>              <math>s = s + i</math>              <math>i = i + 1</math></p>
---	---

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

Opgave 14 (Analyse af løkker, 6%)

<p><b>Algoritme loop1(<math>n</math>)</b>  <math>s = 0</math>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <b>for</b> <math>j = 1</math> <b>to</b> <math>i * i</math>                  <math>s = s + 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq n</math>                  <math>j = j + 1</math>              <math>i = i + 1</math></p>
<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq i</math>                  <math>j = j + 1</math>              <math>i = i + 1</math></p>	<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>s = 1</math>  <b>while</b> <math>s \leq n * n</math>              <math>i = i + 1</math>              <math>s = s + i</math></p>

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

**Opgave 15 (Analyse af løkker, 6%)**

<b>Algoritme loop1(<math>n</math>)</b> $i = n$ <b>while</b> $i > 0$ $i = i - 1$	<b>Algoritme loop2(<math>n</math>)</b> $s = 0$ <b>for</b> $i = 1$ <b>to</b> $n$ <b>for</b> $j = 1$ <b>to</b> $i * i$ $s = s + 1$
<b>Algoritme loop3(<math>n</math>)</b> $s = n$ <b>while</b> $s > 0$ $s = \lfloor s/2 \rfloor$	<b>Algoritme loop4(<math>n</math>)</b> $s = 0$ $i = 1$ <b>while</b> $i * i \leq n$ <b>for</b> $j = 1$ <b>to</b> $i$ $s = s + 1$ $i = i + 1$

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

## Opgave 16 (Analyse af løkker, 6 %)

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

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

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

## Opgave 17 (Analyse af løkker, 6 %)

<b>Algoritme</b> loop1( $n$ )	<b>Algoritme</b> loop2( $n$ )
<b>for</b> $i = 1$ <b>to</b> $n$	$s = 1$
$j = i$	<b>for</b> $i = 1$ <b>to</b> $n$
<b>while</b> $j > 0$	$s = s + 1$
$j = j - 1$	
<b>Algoritme</b> loop3( $n$ )	<b>Algoritme</b> loop4( $n$ )
$i = 0$	$s = 0$
$j = n$	$i = n$
<b>while</b> $i \leq j$	<b>while</b> $i > 1$
$i = i + 1$	<b>for</b> $j = 1$ <b>to</b> $n$
$j = j - 1$	$s = s + 1$
	$i = \lfloor i/2 \rfloor$

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

## Opgave 18 (Analyse af løkker, 6 %)

<b>Algoritme</b> loop1( $n$ ) $s = 0$ <b>for</b> $i = 1$ <b>to</b> $n$ <b>for</b> $j = 1$ <b>to</b> $n$ <b>for</b> $k = 1$ <b>to</b> $n$ $s = s + 1$	<b>Algoritme</b> loop2( $n$ ) $s = 1$ <b>for</b> $i = 1$ <b>to</b> $n$ $s = s + 1$
<b>Algoritme</b> loop3( $n$ ) $i = 1$ $j = n$ <b>while</b> $i \leq j$ $i = i * 2$ $j = \lfloor j/2 \rfloor$	<b>Algoritme</b> loop4( $n$ ) <b>for</b> $i = 1$ <b>to</b> $n$ $j = 1$ <b>while</b> $j \leq i$ $j = j + 1$

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

Opgave 19 (Analyse af løkker, 6 %)

<p><b>Algoritme loop1(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = i</math>              <b>while</b> <math>j &gt; 0</math>                  <math>j = j - 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = n</math>              <b>while</b> <math>i &gt; 0</math>                  <math>i = i - 1</math></p>
<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n * n</math>                  <math>i = 3 * i</math></p>	<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n</math>                  <math>j = 1</math>                  <b>while</b> <math>j \leq i</math>                      <math>j = j + 1</math>                  <math>i = 2 * i</math></p>

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4



## Opgave 20 (Analyse af løkker, 6 %)

**Algoritme loop1( $n$ )**

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

**Algoritme loop3( $n$ )**

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

**Algoritme loop2( $n$ )**

```
for i = 1 to n
  j = i
  while j > 0
    j = j - 1
```

**Algoritme loop4( $n$ )**

```
i = 0
s = 0
q = 0
while q ≤ n
  i = i + 1
  s = s + i
  q = q + s
```

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

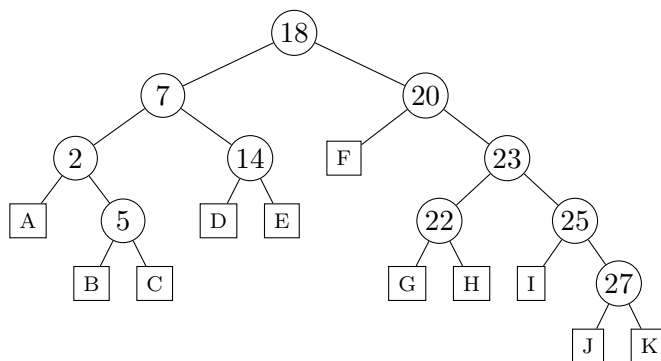
loop1

loop2

loop3

loop4

**Opgave 21 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 6, 21, 4, 10 og 19 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(6)

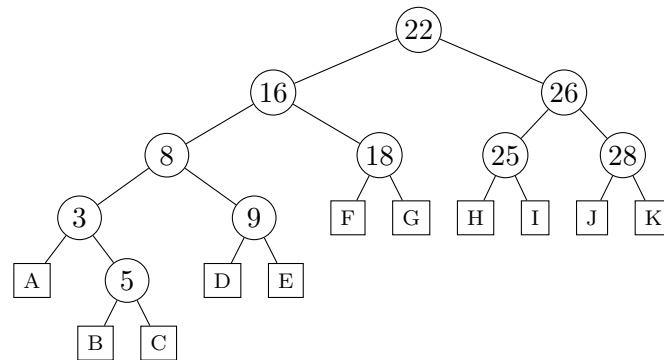
INSERT(21)

INSERT(4)

INSERT(10)

INSERT(19)

**Opgave 22 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 29, 7, 15, 10 og 20 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(29)

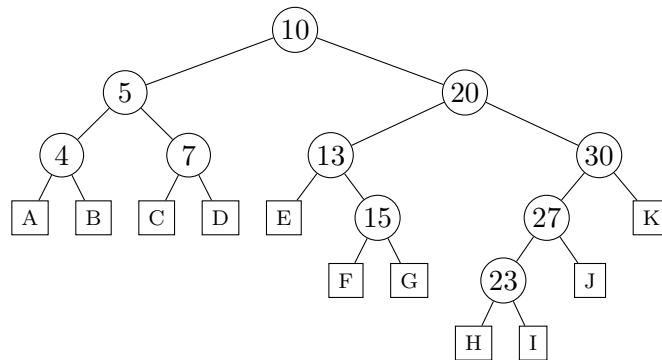
INSERT(7)

INSERT(15)

INSERT(10)

INSERT(20)

**Opgave 23 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 9, 24, 16, 17 og 14 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(9)

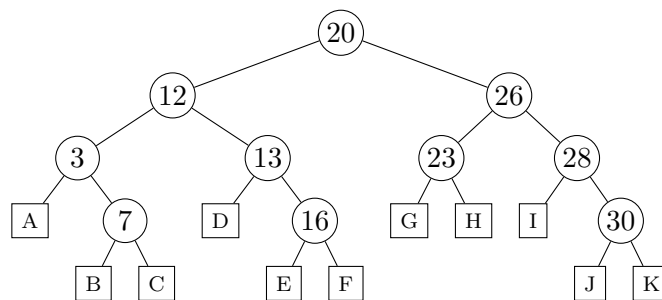
INSERT(24)

INSERT(16)

INSERT(17)

INSERT(14)

**Opgave 24 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 2, 14, 9, 21 og 6 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(2)

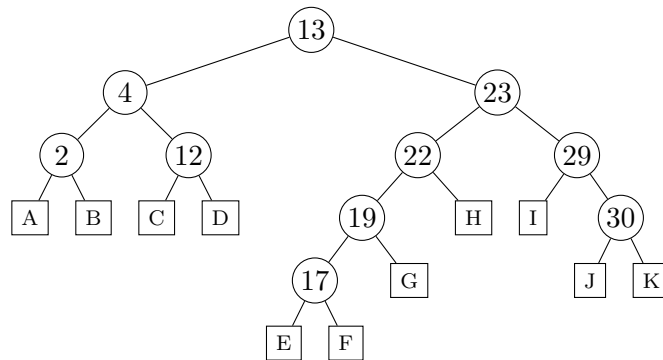
INSERT(14)

INSERT(9)

INSERT(21)

INSERT(6)

**Opgave 25 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 16, 25, 3, 31 og 24 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)

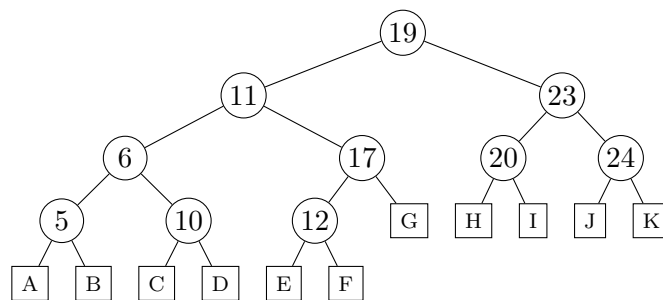
INSERT(25)

INSERT(3)

INSERT(31)

INSERT(24)

**Opgave 26 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 25, 18, 9, 13 og 4 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(25)

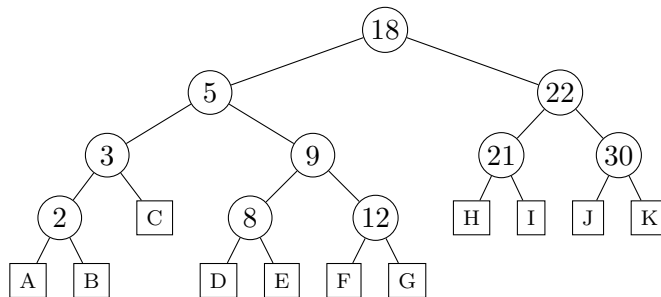
INSERT(18)

INSERT(9)

INSERT(13)

INSERT(4)

**Opgave 27 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 20, 14, 15, 24 og 25 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(20)

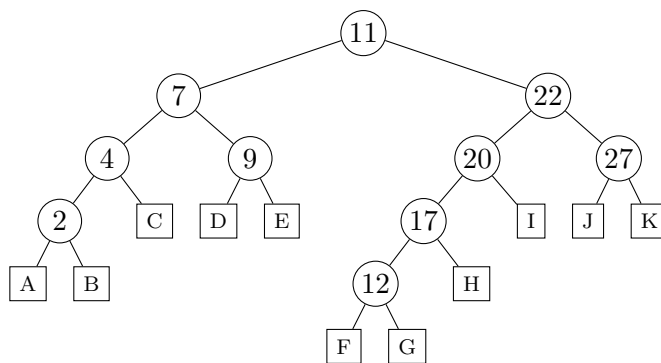
INSERT(14)

INSERT(15)

INSERT(24)

INSERT(25)

**Opgave 28 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 10, 23, 25, 5 og 28 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(10)

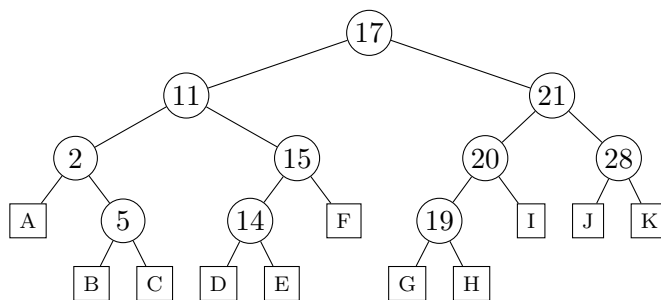
INSERT(23)

INSERT(25)

INSERT(5)

INSERT(28)

**Opgave 29 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 16, 18, 27, 6 og 29 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(16)

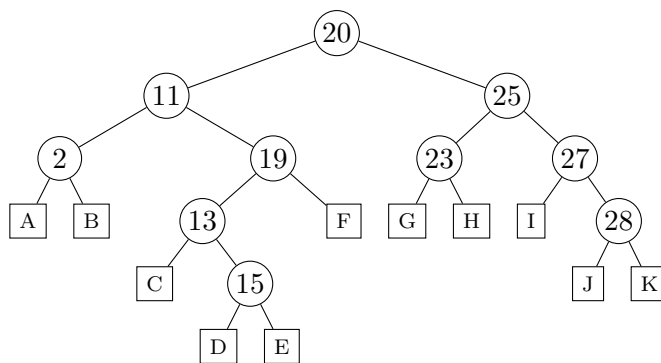
INSERT(18)

INSERT(27)

INSERT(6)

INSERT(29)

**Opgave 30 (Indsættelser i søgetræer, 4 %)**



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 12, 1, 3, 4 og 16 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

A B C D E F G H I J K

INSERT(12)

INSERT(1)

INSERT(3)

INSERT(4)

INSERT(16)

**Opgave 31 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 3, 5, 4, 14, 7, 13 og 12 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 32 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 2, 6, 7, 9, 5, 8 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 33 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 13, 9, 4, 10, 14, 5 og 6 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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



**Opgave 34 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 1, 7, 11, 5, 4, 14 og 6 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 35 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 7, 10, 3, 4, 5, 6 og 11 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 36 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 6, 8, 9, 5, 13, 2 og 10 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 37 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 8, 4, 5, 11, 12, 10 og 9 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 38 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 7, 4, 3, 10, 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	12	7	10	4	3	6

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

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

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

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

**Opgave 39 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 11, 5, 1, 12, 13, 8 og 6 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7
13	12	8	11	5	1	6

1	2	3	4	5	6	7
11	5	1	12	13	8	6

1	2	3	4	5	6	7
11	13	8	12	5	1	6

1	2	3	4	5	6	7
13	12	11	8	6	5	1

1	2	3	4	5	6	7
13	12	8	5	11	1	6

**Opgave 40 (Max-Heap-Insert, 4 %)**

Angiv den binære max-heap efter indsættelse af elementerne 7, 1, 12, 14, 3, 9 og 2 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

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

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

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

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

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

**Opgave 41 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 42 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 43 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 44 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 45 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 46 (Build-Max-Heap, 4 %)**

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

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

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

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

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

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

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

**Opgave 47 (Build-Max-Heap, 4 %)**

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

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

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

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

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

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

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

**Opgave 48 (Build-Max-Heap, 4 %)**

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

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

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

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

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

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

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

**Opgave 49 (Build-Max-Heap, 4 %)**

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

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

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

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

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

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

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

**Opgave 50 (Build-Max-Heap, 4 %)**

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

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

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

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

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

**Opgave 51 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	23	14	17	21	20	3	10	12	5	16	13

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

1	2	3	4	5	6	7	8	9	10	11	12	13
24	17	23	14	12	21	20	3	10		5	16	13

1	2	3	4	5	6	7	8	9	10	11	12
24	17	23	14	13	21	20	3	10	12	5	16

1	2	3	4	5	6	7	8	9	10	11	12
24	23	20	17	21	14	3	10	12	5	16	13

1	2	3	4	5	6	7	8	9	10	11	12
24	17	23	14	12	21	20	3	10	13	5	16

1	2	3	4	5	6	7	8	9	10	11	12
24	17	23	14	12	21	20	3	10	5	16	13



**Opgave 52 (Heap-Extract-Max, 4 %)**

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

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

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

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

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

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

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

**Opgave 53 (Heap-Extract-Max, 4 %)**

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

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

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

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

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

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

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

**Opgave 54 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	23	16	13	18	15	4	11	7	2	8	12

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

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

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

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

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

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

**Opgave 55 (Heap-Extract-Max, 4 %)**

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

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

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

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

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

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

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

**Opgave 56 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
25	21	17	20	18	14	1	5	2	6	16	3	9

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

1	2	3	4	5	6	7	8	9	10	11	12	13
21	20	17	5	18	14	1		2	6	16	3	9

1	2	3	4	5	6	7	8	9	10	11	12
21	20	17	9	18	14	1	5	2	6	16	3

1	2	3	4	5	6	7	8	9	10	11	12
21	18	20	17	16	9	5	2	6	14	3	1

1	2	3	4	5	6	7	8	9	10	11	12
21	20	17	5	18	14	1	9	2	6	16	3

1	2	3	4	5	6	7	8	9	10	11	12
21	20	17	5	18	14	1	2	6	16	3	9

**Opgave 57 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	22	10	14	19	21	7	1	4	3	12	5

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

1	2	3	4	5	6	7	8	9	10	11	12
25	14	22	10	4	19	21	7	1	3	12	5

1	2	3	4	5	6	7	8	9	10	11	12
25	14	22	10	5	19	21	7	1	4	3	12

1	2	3	4	5	6	7	8	9	10	11	12
25	22	21	14	19	10	7	1	4	3	12	5

1	2	3	4	5	6	7	8	9	10	11	12
25	14	22	10	4	19	21	7	1	5	3	12

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

**Opgave 58 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
26	23	19	17	12	15	16	10	9	6	8	1	13

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

1	2	3	4	5	6	7	8	9	10	11	12
23	17	19	10	12	15	16	13	9	6	8	1

1	2	3	4	5	6	7	8	9	10	11	12
23	19	17	12	15	16	10	9	6	8	1	13

1	2	3	4	5	6	7	8	9	10	11	12
23	17	19	10	12	15	16	9	6	8	1	13

1	2	3	4	5	6	7	8	9	10	11	12	13
23	17	19	10	12	15	16		9	6	8	1	13

1	2	3	4	5	6	7	8	9	10	11	12
23	17	19	13	12	15	16	10	9	6	8	1

**Opgave 59 (Heap-Extract-Max, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	24	20	11	21	17	3	1	5	4	15	6

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

1	2	3	4	5	6	7	8	9	10	11	12
25	20	24	3	11	21	17	1	5	4	15	6

1	2	3	4	5	6	7	8	9	10	11	12	13
25	20	24	3	11	21	17		1	5	4	15	6

1	2	3	4	5	6	7	8	9	10	11	12
25	24	20	11	21	17	3	1	5	4	15	6

1	2	3	4	5	6	7	8	9	10	11	12
25	20	24	3	11	21	17	6	1	5	4	15

1	2	3	4	5	6	7	8	9	10	11	12
25	20	24	6	11	21	17	3	1	5	4	15

**Opgave 60 (Heap-Extract-Max, 4 %)**

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

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

1	2	3	4	5	6	7	8	9	10	11	12
25	23	18	15	22	14	11	5	7	3	17	12

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

1	2	3	4	5	6	7	8	9	10	11	12
25	15	23	12	7	22	14	11	5	3	17	18

1	2	3	4	5	6	7	8	9	10	11	12
25	18	23	12	15	22	14	11	5	7	3	17

1	2	3	4	5	6	7	8	9	10	11	12
25	15	23	12	7	22	14	11	5	18	3	17

**Opgave 61 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	9	13	5	29	23	19	12	16	10	25	27	20	17	22

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
24	9	13	5	10	12	16	19	20	23	25	27	29	17	22

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	9	10	12	13	16	17	19	20	22	23	24	25	27	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	9	13	5	19	12	16	10	20	29	23	25	27	17	22

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	9	13	5	19	12	16	10	20	23	25	27	29	17	22

**Opgave 62 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	1	26	5	23	22	29	2	19	27	21	20	3	7	14

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	1	2	3	26	5	23	22	29	19	27	21	20	7	14

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	5	7	14	19	20	21	22	23	26	27	29	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	1	2	3	23	22	29	26	19	27	21	20	5	7	14

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	1	2	3	5	19	20	21	22	23	26	27	29	7	14

**Opgave 63 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	30	24	9	21	19	26	27	8	5	23	22	20	7	2

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	7	8	9	16	19	20	21	22	23	24	26	27	30

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	30	9	19	8	5	20	27	21	24	23	22	26	7	2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	30	9	19	8	5	20	24	21	26	27	23	22	7	2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	30	5	8	9	19	20	21	22	23	24	26	27	7	2

**Opgave 64 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	19	29	20	2	26	4	11	12	5	24	9	21	7	1

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
13	19	2	4	5	7	9	11	12	20	21	24	26	29	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	19	2	4	5	7	20	11	12	29	24	9	21	26	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	19	2	4	5	7	29	20	26	11	12	24	9	21	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	4	5	7	9	11	12	13	19	20	21	24	26	29

**Opgave 65 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
22	2	10	3	23	20	29	11	25	18	13	14	19	26	6

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	6	10	11	13	14	18	19	20	22	23	25	26	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
22	2	10	3	11	13	14	23	25	18	20	29	19	26	6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
22	2	3	10	11	13	14	18	20	23	25	29	19	26	6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
22	2	10	3	11	13	14	23	20	29	25	18	19	26	6

**Opgave 66 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	4	23	7	13	14	25	2	16	18	17	20	12	8	29

Angiv resultatet af at anvende PARTITION( $A, 3, 12$ ) på ovenstående array  $A$ .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	7	8	12	13	14	16	17	18	20	23	25	27	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	4	7	13	14	2	16	18	17	20	23	25	12	8	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	4	2	7	13	14	16	17	18	20	23	25	12	8	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	4	7	13	14	2	16	18	17	20	25	23	12	8	29

**Opgave 67 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	2	20	7	28	8	14	25	29	11	1	23	18	19	22

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
21	2	7	8	14	11	1	18	20	28	25	29	23	19	22

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	7	8	11	14	18	19	20	21	22	23	25	28	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	1	2	7	8	11	14	18	20	23	25	28	29	19	22

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	2	7	8	14	11	1	18	29	20	28	23	25	19	22

**Opgave 68 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	5	15	27	28	23	8	20	16	4	18	2	14	13	19

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	8	13	14	15	16	18	19	20	23	24	27	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	2	5	15	27	28	23	8	20	16	4	18	14	13	19

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	2	4	5	8	15	16	18	20	23	27	28	14	13	19

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	2	15	27	28	23	8	20	16	4	18	5	14	13	19

**Opgave 69 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	7	8	1	4	25	27	26	19	12	6	13	24	14	20

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
9	7	1	4	6	8	12	13	14	19	24	25	26	27	20

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	6	7	8	9	12	13	14	19	20	24	25	26	27

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	7	8	1	4	12	6	13	14	25	27	26	19	24	20

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	7	8	1	4	12	6	13	14	25	27	26	24	19	20

**Opgave 70 (Partition, 4 %)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
26	16	7	25	17	20	18	24	19	27	2	28	8	3	9

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	7	8	9	16	17	18	19	20	24	25	26	27	28

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
26	16	7	2	8	25	17	20	18	24	19	27	28	3	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
26	16	2	7	8	17	18	19	20	24	25	27	28	3	9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
26	16	7	2	8	20	18	24	19	27	25	28	17	3	9



**Opgave 71 (Radix-sort, 4 %)**

2113 4313 3413 3404 1112 2313

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

1112 2113 2313 3413 3404 4313  
 3404 1112 2113 2313 3413 4313  
 1112 2113 4313 3413 2313 3404  
 3404 1112 2113 4313 3413 2313  
 1112 2113 2313 3404 3413 4313

**Opgave 72 (Radix-sort, 4 %)**

2433 2303 1230 2430 3103 0330

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

0330 1230 2303 2430 2433 3103  
 2303 3103 0330 1230 2430 2433  
 2303 3103 1230 2430 0330 2433  
 1230 2430 0330 2303 3103 2433  
 0330 1230 2303 2433 2430 3103

**Opgave 73 (Radix-sort, 4 %)**

4021 2321 1222 4022 2313 0122

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

0122 1222 2313 2321 4021 4022  
 0122 1222 2321 2313 4021 4022  
 2313 2321 4021 0122 1222 4022  
 4021 2321 1222 4022 0122 2313  
 2313 4021 2321 1222 4022 0122

**Opgave 74 (Radix-sort, 4 %)**

1400 2000 3120 3111 2120 0220

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

1400 2000 3111 0220 2120 3120

1400 2000 3111 3120 2120 0220

0220 1400 2000 2120 3111 3120

1400 2000 3120 2120 0220 3111

0220 1400 2000 2120 3120 3111

**Opgave 75 (Radix-sort, 4 %)**

2242 1313 0442 1034 2234 1134

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

0442 1034 1134 1313 2242 2234

2242 0442 1313 1034 2234 1134

1313 1034 1134 2234 0442 2242

1313 1034 2234 1134 2242 0442

0442 1034 1134 1313 2234 2242

**Opgave 76 (Radix-sort, 4 %)**

4210 2232 0101 0132 3432 2201

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

0101 0132 2201 2232 3432 4210

0101 2201 4210 2232 0132 3432

0101 0132 2232 2201 3432 4210

0101 2201 4210 0132 2232 3432

4210 0101 2201 2232 0132 3432

**Opgave 77 (Radix-sort, 4 %)**

0431 4331 3231 0420 3014 0320

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

3014 0420 0320 0431 4331 3231  
 0320 0431 0420 3014 3231 4331  
 0420 0320 0431 4331 3231 3014  
 0320 0420 0431 3014 3231 4331  
 3014 0320 0420 0431 3231 4331

**Opgave 78 (Radix-sort, 4 %)**

0044 4421 0003 3121 3044 1303

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

0003 1303 3121 4421 0044 3044  
 0044 0003 1303 3044 3121 4421  
 0003 0044 1303 3044 3121 4421  
 4421 3121 0003 1303 0044 3044  
 0003 1303 4421 3121 0044 3044

**Opgave 79 (Radix-sort, 4 %)**

4131 1012 4034 0312 4234 4212

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

0312 1012 4034 4131 4212 4234  
 1012 0312 4212 4131 4034 4234  
 4131 1012 0312 4212 4034 4234  
 0312 1012 4034 4131 4234 4212  
 0312 1012 4212 4131 4034 4234

**Opgave 80 (Radix-sort, 4 %)**

1241 4140 4231 0240 1131 1240

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

4140 0240 1240 4231 1131 1241  
 0240 1131 1240 1241 4140 4231  
 4231 1131 4140 0240 1240 1241  
 0240 1131 1241 1240 4140 4231  
 1131 4231 0240 1240 4140 1241

**Opgave 81 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
14		19						21	3	18

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

Angiv positionerne de fem elementer 0, 1, 2, 4 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 14, 18, 19 og 21).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(1)

INSERT(2)

INSERT(4)

INSERT(7)

**Opgave 82 (Lineær probing, 4%)**

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

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

Angiv positionerne de fem elementer 0, 4, 8, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 5, 7, 11 og 16).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(4)

INSERT(8)

INSERT(9)

INSERT(10)

**Opgave 83 (Lineær probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
		6	20	1	17					19

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

Angiv positionerne de fem elementer 3, 8, 9, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 6, 17, 19 og 20).

0 1 2 3 4 5 6 7 8 9 10

INSERT(3)

INSERT(8)

INSERT(9)

INSERT(10)

INSERT(11)

**Opgave 84 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
22			12					10	14	3

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

Angiv positionerne de fem elementer 2, 6, 7, 8 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 10, 12, 14 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(6)

INSERT(7)

INSERT(8)

INSERT(11)

**Opgave 85 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
11							19	8	15	13

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

Angiv positionerne de fem elementer 2, 6, 7, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 11, 13, 15 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(6)

INSERT(7)

INSERT(9)

INSERT(10)

**Opgave 86 (Lineær probing, 4 %)**

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

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

Angiv positionerne de fem elementer 0, 3, 7, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 6, 16, 18 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(3)

INSERT(7)

INSERT(8)

INSERT(9)

**Opgave 87 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
	15		1					10	21	18

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

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 1, 10, 15, 18 og 21).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(3)

INSERT(5)

INSERT(7)

INSERT(9)

**Opgave 88 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
22	11							13	16	5

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

Angiv positionerne de fem elementer 0, 4, 6, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 5, 11, 13, 16 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(4)

INSERT(6)

INSERT(8)

INSERT(9)

**Opgave 89 (Lineær probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
0	22				20				14	18

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

Angiv positionerne de fem elementer 3, 4, 5, 7 og 8 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 14, 18, 20 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(3)

INSERT(4)

INSERT(5)

INSERT(7)

INSERT(8)



**Opgave 90 (Lineær probing, 4%)**

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

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

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, 9, 12 og 19).

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

INSERT(2)

INSERT(6)

INSERT(7)

INSERT(8)

INSERT(10)

**Opgave 91 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
			5		4		21	13	16	

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

Angiv positionerne de fem elementer 2, 6, 8, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 5, 13, 16 og 21).

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

INSERT(2)

INSERT(6)

INSERT(8)

INSERT(9)

INSERT(11)

**Opgave 92 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
1		12	18				9	15		

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

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 1, 9, 12, 15 og 18).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(4)

INSERT(5)

INSERT(7)

INSERT(8)

**Opgave 93 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
22		11				14	20		9	

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 2, 3, 4, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 9, 11, 14, 20 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(3)

INSERT(4)

INSERT(8)

INSERT(10)

**Opgave 94 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
0	20		16			21			9	

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

Angiv positionerne de fem elementer 1, 2, 3, 4 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 9, 16, 20 og 21).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(2)

INSERT(3)

INSERT(4)

INSERT(11)

**Opgave 95 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
		18	6	14				17		2

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

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 2, 6, 14, 17 og 18).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(3)

INSERT(4)

INSERT(5)

INSERT(9)

**Opgave 96 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
	3			19	15			13		4

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

Angiv positionerne de fem elementer 1, 2, 5, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 4, 13, 15 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(2)

INSERT(5)

INSERT(10)

INSERT(11)

**Opgave 97 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
22			7				20	11		16

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

Angiv positionerne de fem elementer 0, 1, 4, 6 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 7, 11, 16, 20 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(1)

INSERT(4)

INSERT(6)

INSERT(10)

**Opgave 98 (Kvadratisk probing, 4%)**

0	1	2	3	4	5	6	7	8	9	10
11	14			19			8	22		

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

Angiv positionerne de fem elementer 0, 2, 6, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 11, 14, 19 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(2)

INSERT(6)

INSERT(9)

INSERT(10)

**Opgave 99 (Kvadratisk probing, 4%)**

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

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

Angiv positionerne de fem elementer 4, 5, 6, 8 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 7, 14, 16 og 20).

0 1 2 3 4 5 6 7 8 9 10

INSERT(4)

INSERT(5)

INSERT(6)

INSERT(8)

INSERT(11)

**Opgave 100 (Kvadratisk probing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
		19	12		9		17	1		

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

Angiv positionerne de fem elementer 0, 3, 6, 8 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 9, 12, 17 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(0)

INSERT(3)

INSERT(6)

INSERT(8)

INSERT(10)

**Opgave 101 (Dobbelt hashing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
0		18	20			7			16	

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

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 0, 7, 16, 18 og 20).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(4)

INSERT(6)

INSERT(9)

INSERT(10)

**Opgave 102 (Dobbelt hashing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
6				3			19	17	4	

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

Angiv positionerne de fem elementer 5, 7, 8, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 3, 4, 6, 17 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(5)

INSERT(7)

INSERT(8)

INSERT(10)

INSERT(11)

**Opgave 103 (Dobbelt hashing, 4 %)**

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

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 4, 6, 8, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 2, 13, 15 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(4)

INSERT(6)

INSERT(8)

INSERT(10)

INSERT(11)

**Opgave 104 (Dobbelt hashing, 4 %)**

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

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne  $h_1(k) = 5k \bmod 11$  og  $h_2(k) = 1 + (3k \bmod 10)$ .

Angiv positionerne de fem elementer 1, 3, 7, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 4, 12, 13 og 15).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(3)

INSERT(7)

INSERT(9)

INSERT(11)

**Opgave 105 (Dobbelt hashing, 4 %)**

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

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

Angiv positionerne de fem elementer 1, 2, 3, 5 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 6, 11, 18 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(2)

INSERT(3)

INSERT(5)

INSERT(7)



**Opgave 106 (Dobbelt hashing, 4 %)**

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

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne  $h_1(k) = 5k \bmod 11$  og  $h_2(k) = 1 + (3k \bmod 10)$ .

Angiv positionerne de fem elementer 1, 3, 4, 10 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 5, 7, 8 og 19).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(3)

INSERT(4)

INSERT(10)

INSERT(11)

**Opgave 107 (Dobbelt hashing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
	14	12		1		18	21			

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

Angiv positionerne de fem elementer 4, 5, 7, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 12, 14, 18 og 21).

0 1 2 3 4 5 6 7 8 9 10

INSERT(4)

INSERT(5)

INSERT(7)

INSERT(9)

INSERT(10)

**Opgave 108 (Dobbelt hashing, 4 %)**

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

I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne  $h_1(k) = 5k \bmod 11$  og  $h_2(k) = 1 + (3k \bmod 10)$ .

Angiv positionerne de fem elementer 2, 5, 7, 8 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 4, 6, 15 og 17).

0 1 2 3 4 5 6 7 8 9 10

INSERT(2)

INSERT(5)

INSERT(7)

INSERT(8)

INSERT(11)

**Opgave 109 (Dobbelt hashing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
22	0		18	13						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 3, 4, 5, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 13, 16, 18 og 22).

0 1 2 3 4 5 6 7 8 9 10

INSERT(3)

INSERT(4)

INSERT(5)

INSERT(9)

INSERT(11)

**Opgave 110 (Dobbelt hashing, 4 %)**

0	1	2	3	4	5	6	7	8	9	10
11	0			5	20		17			

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

Angiv positionerne de fem elementer 1, 2, 6, 9 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 5, 11, 17 og 20).

0 1 2 3 4 5 6 7 8 9 10

INSERT(1)

INSERT(2)

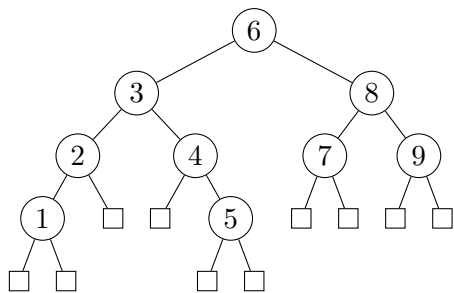
INSERT(6)

INSERT(9)

INSERT(10)

**Opgave 111 (Rød-sort træ, 4 %)**

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



Ja Nej

1, 5

1, 2, 4, 5, 7, 9

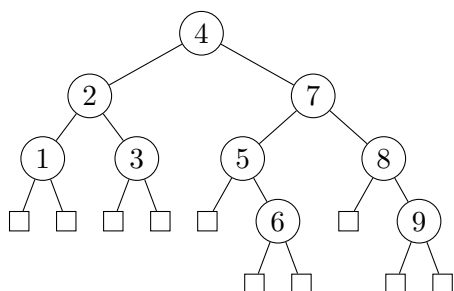
1, 3, 5, 8

1, 3, 5, 7, 9

1, 5, 6

**Opgave 112 (Rød-sort træ, 4 %)**

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



Ja Nej

1, 3, 6, 7, 9

4, 6, 9

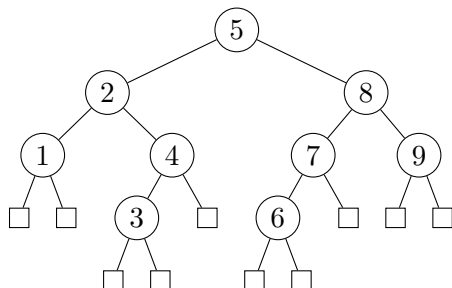
2, 5, 6, 8, 9

6, 9

2, 6, 7, 9

**Opgave 113 (Rød-sort træ, 4%)**

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



Ja Nej

1, 3, 4, 6, 8

3, 6

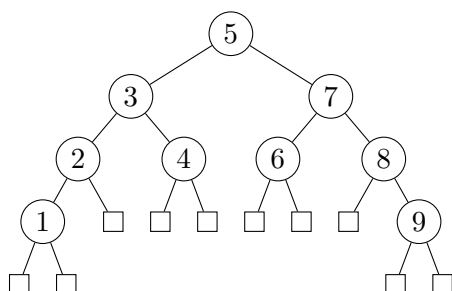
2, 3, 6, 7, 9

2, 3, 6, 8

3, 5, 6

**Opgave 114 (Rød-sort træ, 4%)**

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



Ja Nej

1, 5, 9

1, 3, 6, 8, 9

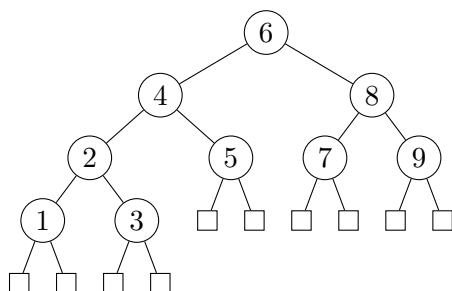
1, 3, 7, 9

1, 2, 4, 7, 9

1, 9

**Opgave 115 (Rød-sort træ, 4%)**

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



Ja Nej

1, 3, 6

1, 3, 4, 7, 9

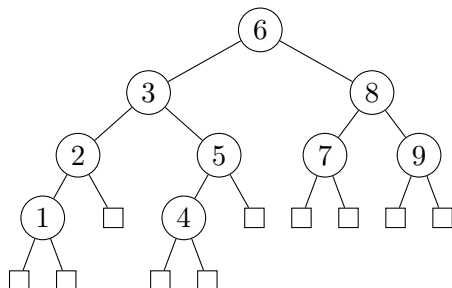
1, 3, 4, 8

1, 3

1, 2, 3, 5, 8

**Opgave 116 (Rød-sort træ, 4%)**

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



Ja Nej

1, 2, 4, 5, 8

1, 4, 6

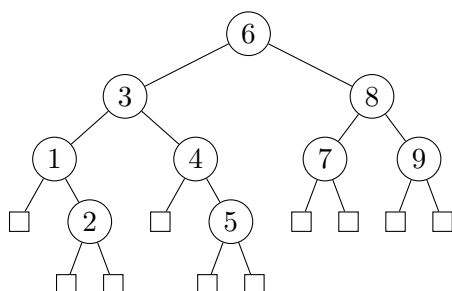
1, 4

1, 3, 4, 7, 9

1, 3, 4, 8

**Opgave 117 (Rød-sort træ, 4%)**

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



Ja Nej

1, 2, 4, 5, 8

2, 5, 6

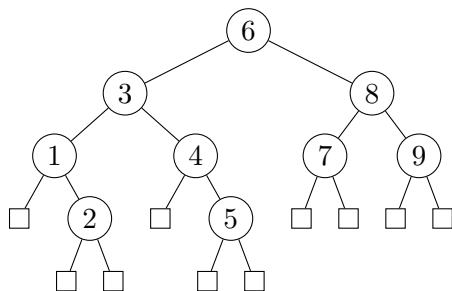
2, 3, 5, 8

2, 5

2, 3, 5, 7, 9

**Opgave 118 (Rød-sort træ, 4%)**

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



Ja Nej

2, 3, 5, 7, 9

1, 2, 4, 5, 8

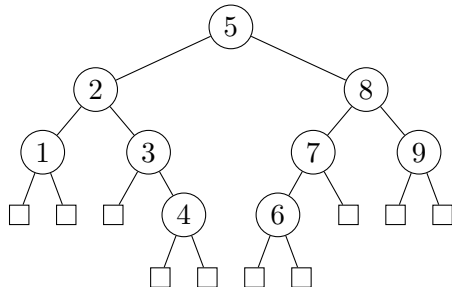
2, 3, 5, 8

2, 5

2, 5, 6

**Opgave 119 (Rød-sort træ, 4%)**

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



Ja Nej

2, 4, 6, 8

2, 4, 6, 7, 9

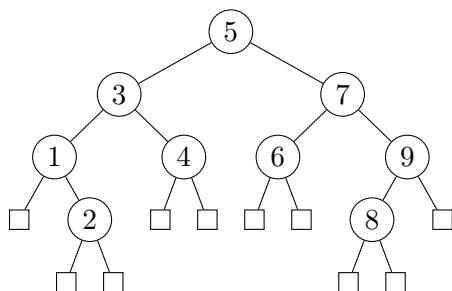
4, 6

4, 5, 6

1, 3, 4, 6, 8

**Opgave 120 (Rød-sort træ, 4%)**

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



Ja Nej

2, 3, 6, 8, 9

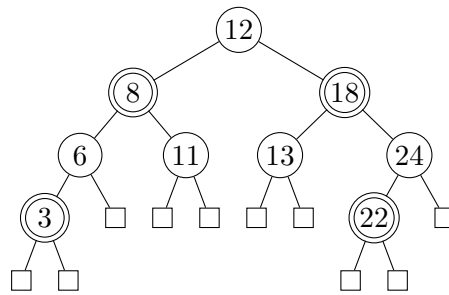
1, 2, 4, 7, 8

2, 8

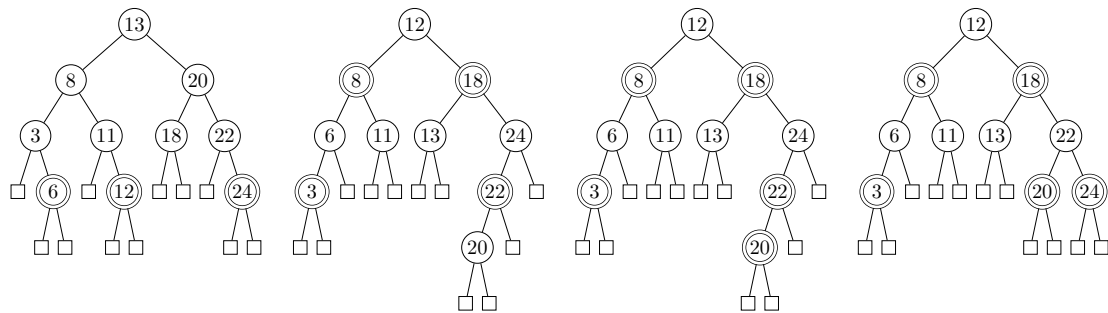
2, 5, 8

2, 3, 7, 8

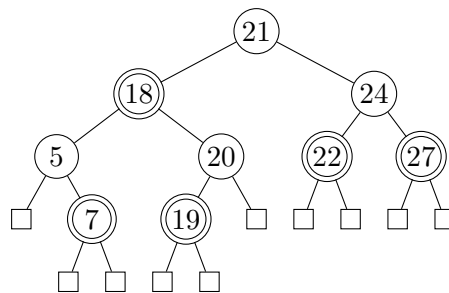
**Opgave 121 (Indsættelse i rød-sort træer, 4 %)**



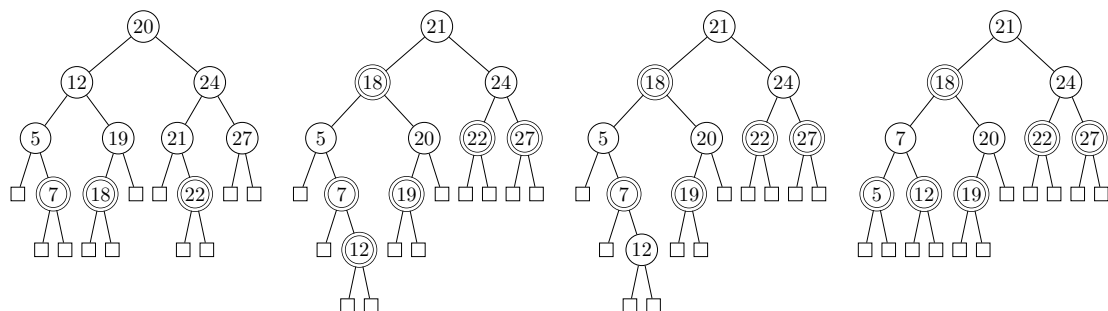
Angiv det resulterende rød-sortede træ når man indsætter 20 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



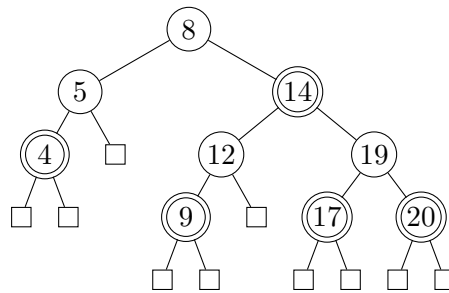
**Opgave 122 (Indsættelse i rød-sort træer, 4 %)**



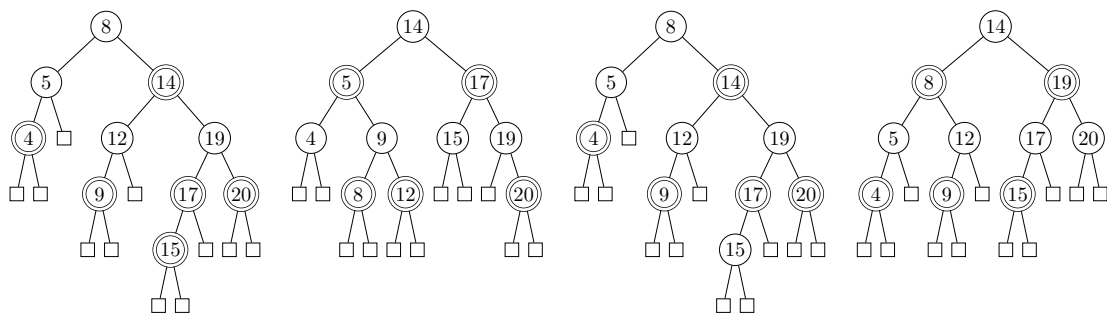
Angiv det resulterende rød-sortede træ når man indsætter 12 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



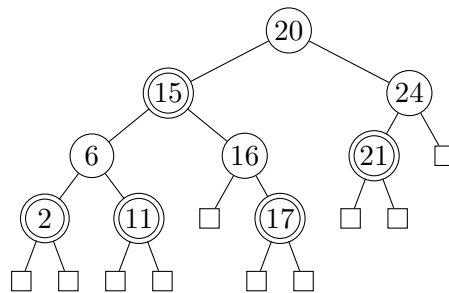
**Opgave 123 (Indsættelse i rød-sort træer, 4 %)**



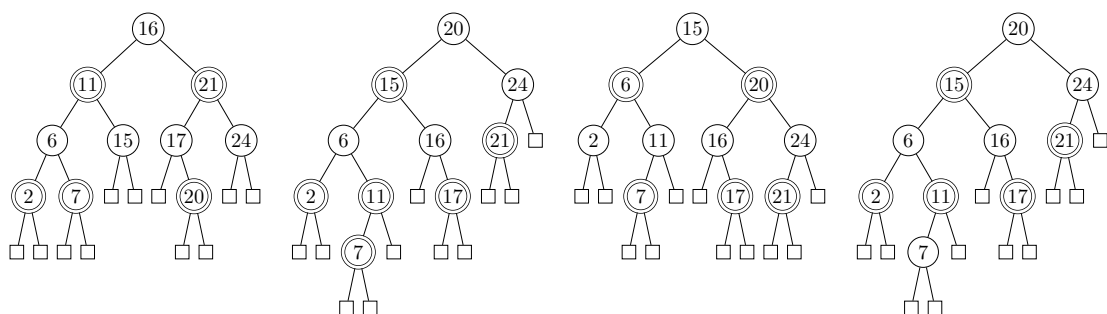
Angiv det resulterende rød-sortede træ når man indsætter 15 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



**Opgave 124 (Indsættelse i rød-sort træer, 4 %)**

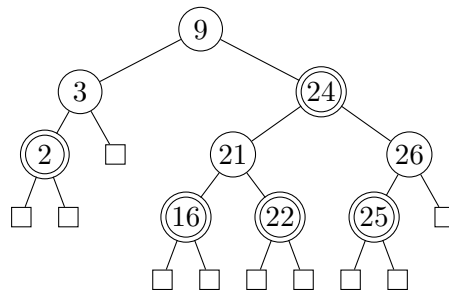


Angiv det resulterende rød-sortede træ når man indsætter 7 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).

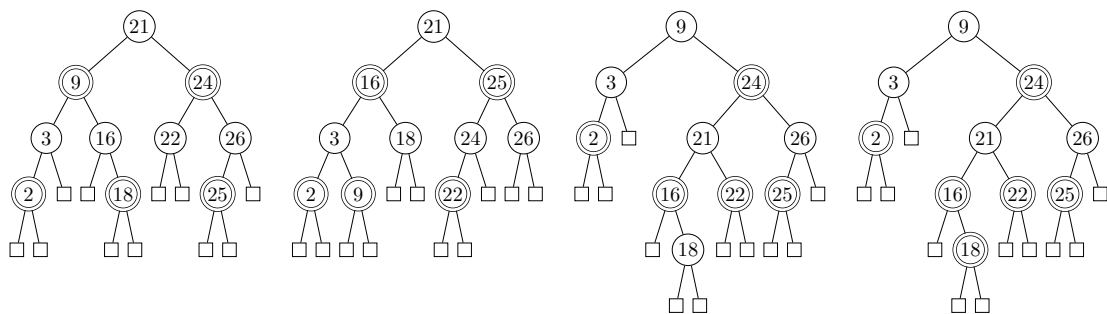




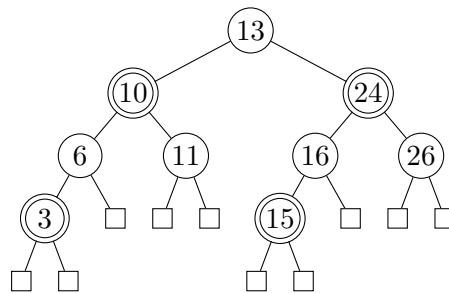
**Opgave 125 (Indsættelse i rød-sort træer, 4 %)**



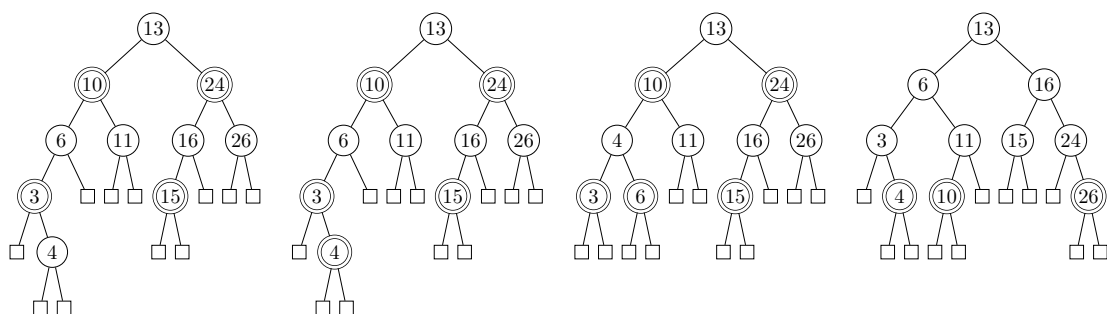
Angiv det resulterende rød-sort træ når man indsætter 18 i ovenstående rød-sort træ (dobbeltcirkler angiver røde knuder).



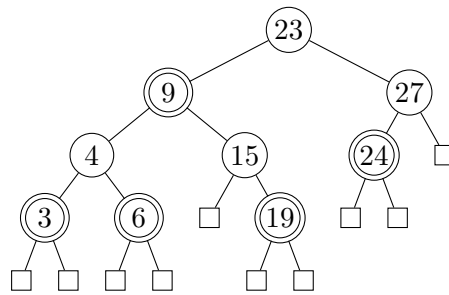
**Opgave 126 (Indsættelse i rød-sort træer, 4 %)**



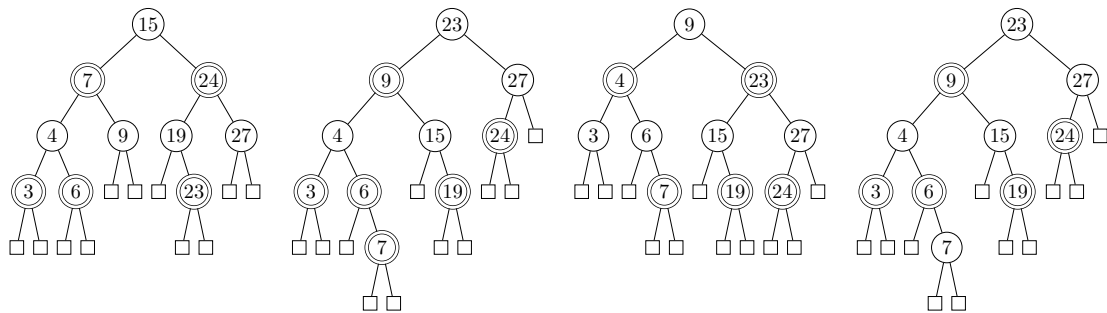
Angiv det resulterende rød-sort træ når man indsætter 4 i ovenstående rød-sort træ (dobbeltcirkler angiver røde knuder).



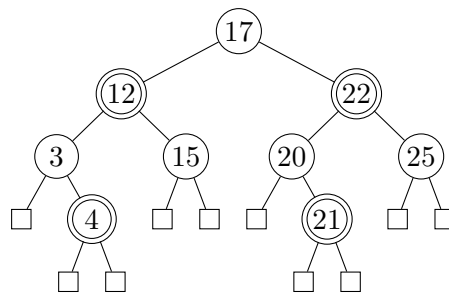
**Opgave 127 (Indsættelse i rød-sort træer, 4 %)**



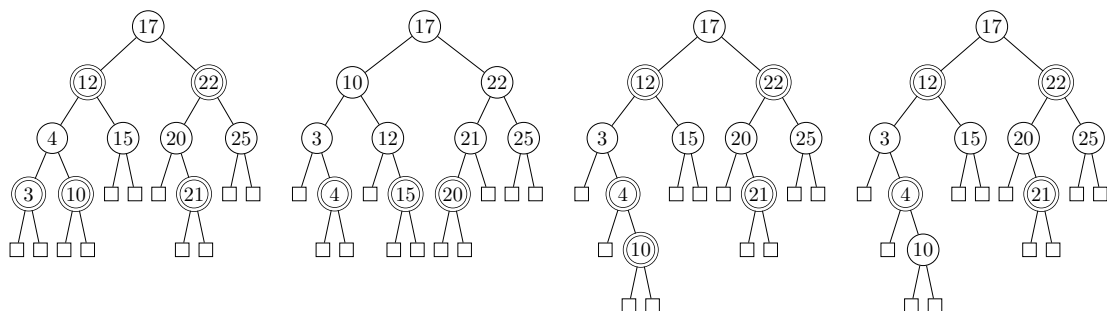
Angiv det resulterende rød-sortede træ når man indsætter 7 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



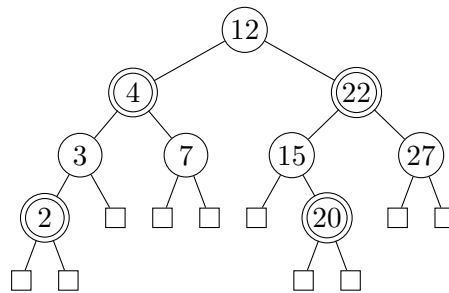
**Opgave 128 (Indsættelse i rød-sort træer, 4 %)**



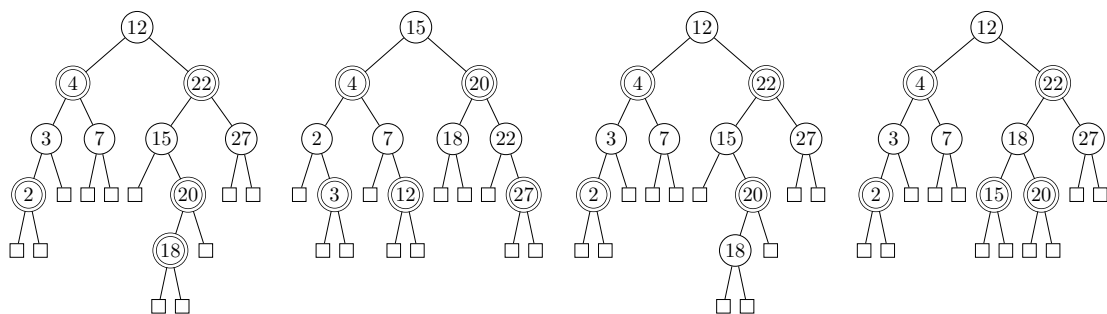
Angiv det resulterende rød-sortede træ når man indsætter 10 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



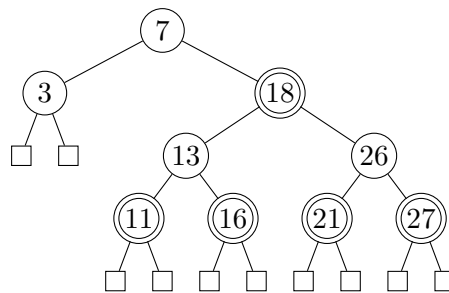
**Opgave 129 (Indsættelse i rød-sort træer, 4 %)**



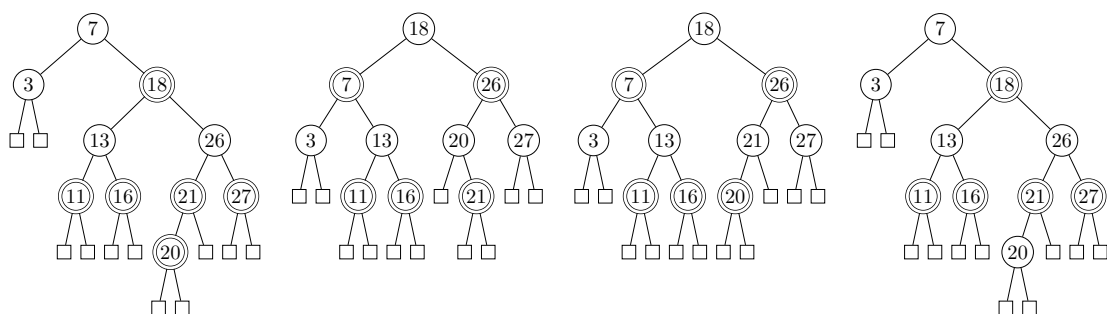
Angiv det resulterende rød-sortede træ når man indsætter 18 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



**Opgave 130 (Indsættelse i rød-sort træer, 4 %)**



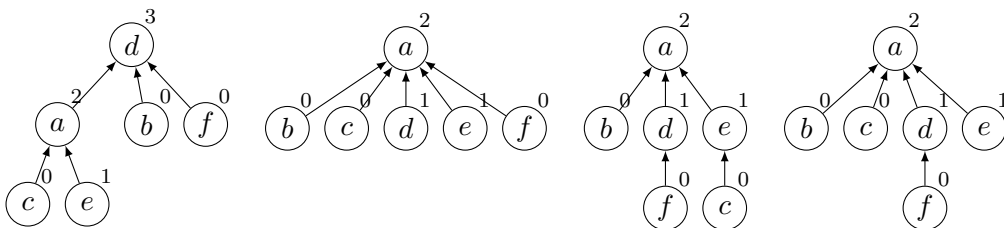
Angiv det resulterende rød-sortede træ når man indsætter 20 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



**Opgave 131 (Union-find, 4 %)**

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

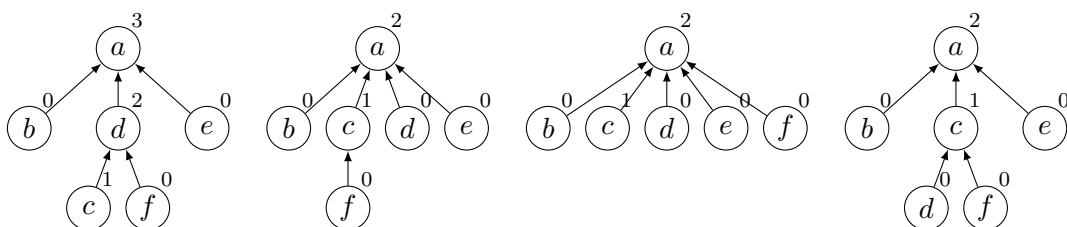
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*c*, *e*)
- UNION(*b*, *a*)
- UNION(*e*, *a*)
- UNION(*f*, *d*)
- UNION(*c*, *f*)
- FIND-SET(*b*)



**Opgave 132 (Union-find, 4 %)**

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

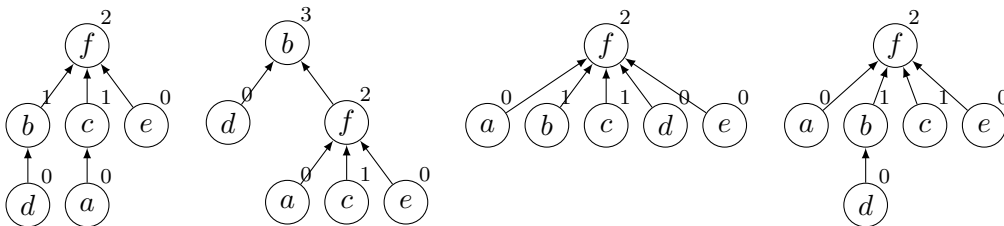
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*f*, *c*)
- UNION(*c*, *d*)
- UNION(*b*, *a*)
- UNION(*f*, *b*)
- UNION(*e*, *d*)
- FIND-SET(*b*)



**Opgave 133 (Union-find, 4 %)**

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

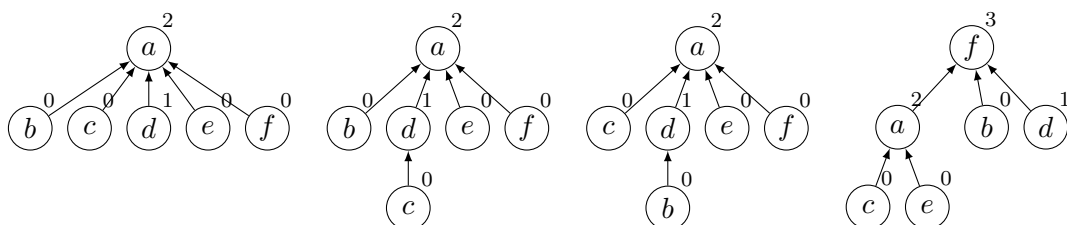
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*a*, *c*)
- UNION(*e*, *f*)
- UNION(*c*, *f*)
- UNION(*d*, *b*)
- UNION(*a*, *d*)
- FIND-SET(*b*)



**Opgave 134 (Union-find, 4 %)**

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

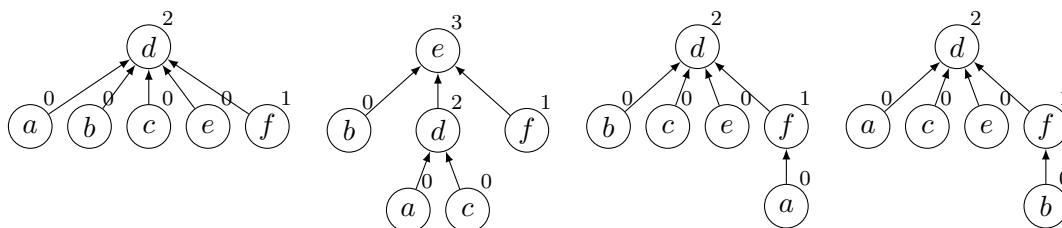
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*c*, *d*)
- UNION(*b*, *c*)
- UNION(*e*, *a*)
- UNION(*d*, *a*)
- UNION(*c*, *f*)
- FIND-SET(*b*)



**Opgave 135 (Union-find, 4 %)**

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

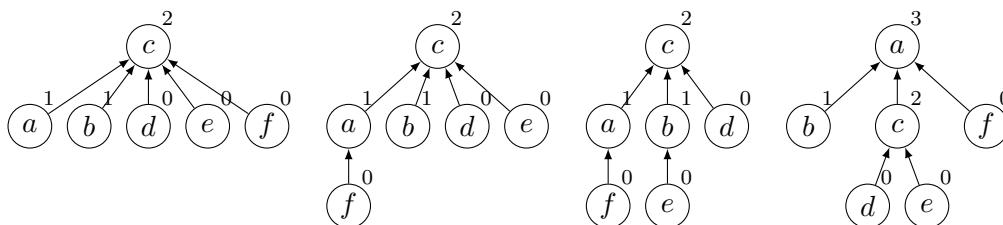
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*b*, *f*)
- UNION(*a*, *b*)
- UNION(*c*, *d*)
- UNION(*f*, *d*)
- UNION(*a*, *e*)
- FIND-SET(*b*)



**Opgave 136 (Union-find, 4 %)**

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

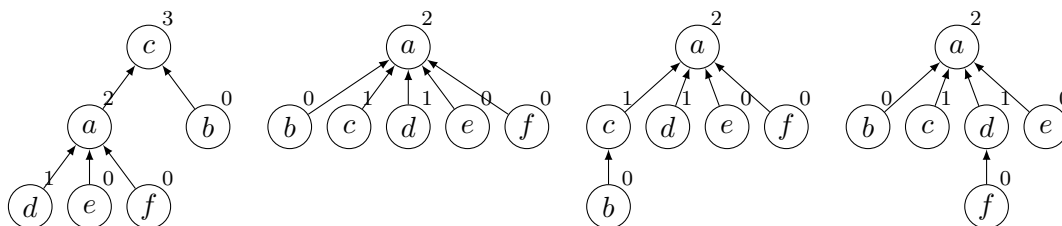
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*e*, *b*)
- UNION(*d*, *c*)
- UNION(*e*, *d*)
- UNION(*f*, *a*)
- UNION(*e*, *f*)
- FIND-SET(*b*)



**Opgave 137 (Union-find, 4 %)**

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

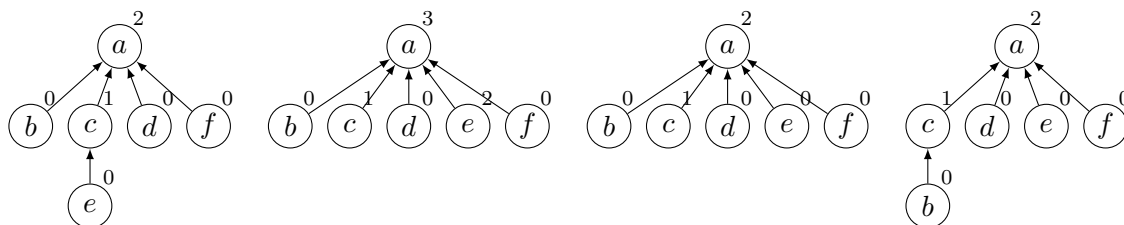
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*f*, *d*)
- UNION(*e*, *a*)
- UNION(*f*, *a*)
- UNION(*b*, *c*)
- UNION(*f*, *c*)
- FIND-SET(*b*)



**Opgave 138 (Union-find, 4 %)**

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

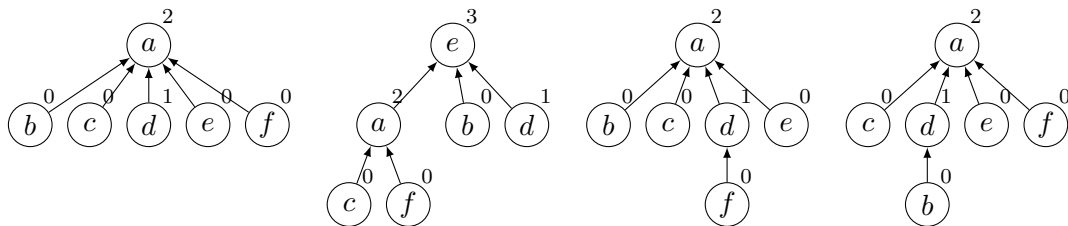
- MAKESET(*a*)
- MAKESET(*b*)
- MAKESET(*c*)
- MAKESET(*d*)
- MAKESET(*e*)
- MAKESET(*f*)
- UNION(*b*, *c*)
- UNION(*b*, *e*)
- UNION(*f*, *a*)
- UNION(*e*, *f*)
- UNION(*d*, *e*)
- FIND-SET(*b*)



**Opgave 139 (Union-find, 4 %)**

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

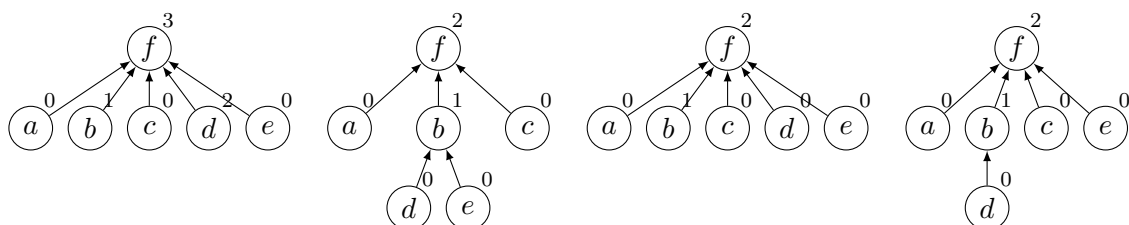
- MAKESET( $a$ )
- MAKESET( $b$ )
- MAKESET( $c$ )
- MAKESET( $d$ )
- MAKESET( $e$ )
- MAKESET( $f$ )
- UNION( $f, d$ )
- UNION( $b, f$ )
- UNION( $c, a$ )
- UNION( $d, a$ )
- UNION( $f, e$ )
- FIND-SET( $b$ )



**Opgave 140 (Union-find, 4 %)**

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





**Opgave 141 (Rekursionsligninger, 4 %)**

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

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

$$T(n) = T(n/4) + 4$$

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

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

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

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

**Opgave 142 (Rekursionsligninger, 4 %)**

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

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

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

$$T(n) = T(n/4) + 5$$

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

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

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

**Opgave 143 (Rekursionsligninger, 4 %)**

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

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

$$T(n) = 8 \cdot T(n/2) + 3$$

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

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

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

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

**Opgave 144 (Rekursionsligninger, 4 %)**

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

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

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

$$T(n) = T(n-1) + 3$$

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

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

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

**Opgave 145 (Rekursionsligninger, 4 %)**

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

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

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

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

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

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

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

**Opgave 146 (Rekursionsligninger, 4 %)**

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

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

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

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

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

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

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

**Opgave 147 (Rekursionsligninger, 4 %)**

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

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

$$T(n) = 8 \cdot T(n/2) + 3$$

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

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

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

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

**Opgave 148 (Rekursionsligninger, 4 %)**

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

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

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

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

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

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

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

**Opgave 149 (Rekursionsligninger, 4 %)**

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

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

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

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

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

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

$$T(n) = T(n/5) + 5$$

**Opgave 150 (Rekursionsligninger, 4 %)**

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

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

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

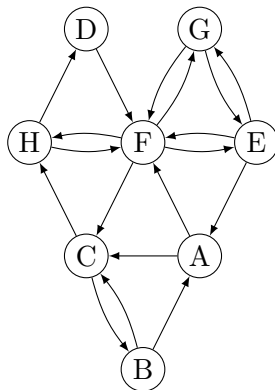
$$T(n) = 4 \cdot T(n/2) + 2$$

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

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

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

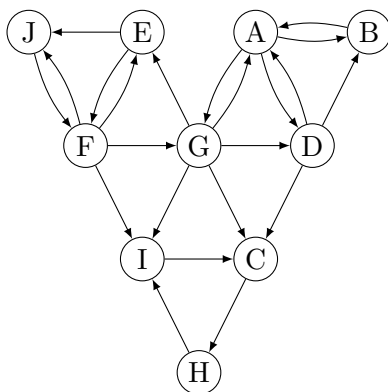
**Opgave 151 (BFS, 4 %)**



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

AFCHGEBD    ACFBHEGD    ACBHDFEG    ACFBHGED

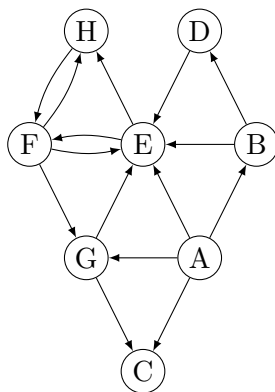
Opgave 152 (BFS, 4 %)



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

ABDGCEIHJF    ABDCHIGEFJ    ABDGCEIHFJ    AGBDCIEHFJ

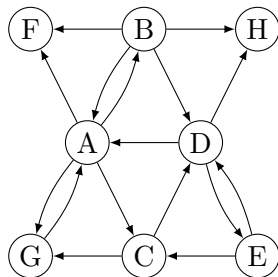
Opgave 153 (BFS, 4 %)



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

ABCEGDFH    ABDEFGCH    ABCEGDHF    AEGBCHFD

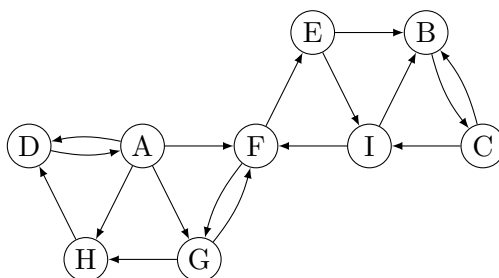
Opgave 154 (BFS, 4 %)



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

- ABCFGDHE    ABCFGHDE    ACBFGDHE    ABDECGHF

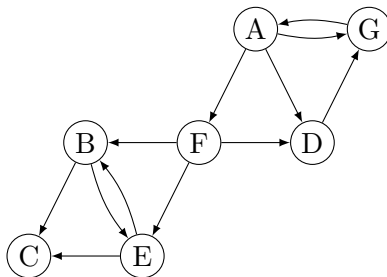
Opgave 155 (BFS, 4 %)



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

- AFDHGEBIC    ADFEBCIGH    ADFGHEBIC    ADFGHEIBC

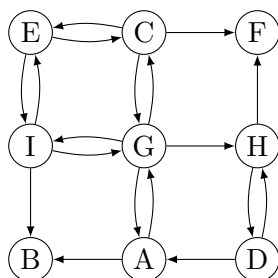
Opgave 156 (BFS, 4 %)



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

- ADFGEB C    ADGFEB C    ADGFBCE    ADFGBEC

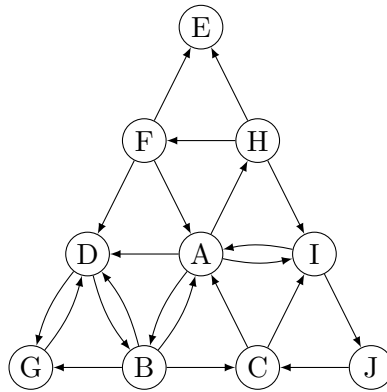
Opgave 157 (BFS, 4 %)



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

- ABGCHIFED    ABGCHIEFD    ABGCEIFHD    ABGICHEFD

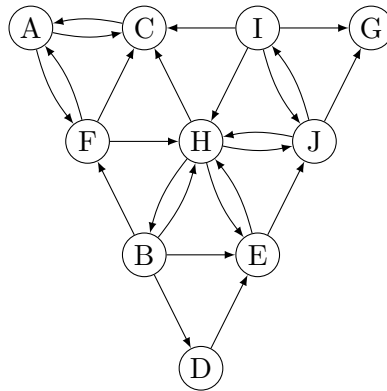
Opgave 158 (BFS, 4 %)



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

AHDBIFEGCJ    ABCIJDGHEF    ABDHICGFEJ    ABDHICGEFJ

Opgave 159 (BFS, 4 %)

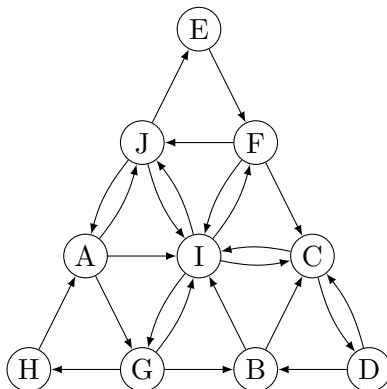


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen  $Q$  i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

ACFHBDEJGI    ACFHBEJDGI    ACFHBEJDIG    ACFHEBJDGI



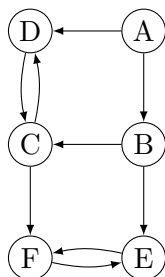
Opgave 160 (BFS, 4 %)



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

AGIJBHCFED    AIGJCFBHED    AGIJBHFCED    AGBCDIFJEH

Opgave 161 (Lovlige bredde først træer, 4 %)



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

Ja    Nej

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

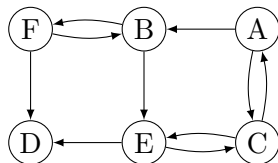
(A,B) (A,D) (B,C) (B,E) (E,F)

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

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

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

**Opgave 162 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

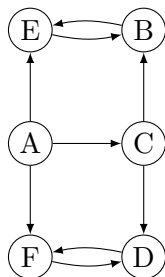
(A,B) (A,C) (B,F) (C,E) (E,D)

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

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

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

**Opgave 163 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

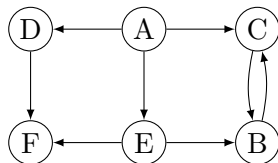
(A,C) (A,E) (C,D) (D,F) (E,B)

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

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

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

**Opgave 164 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

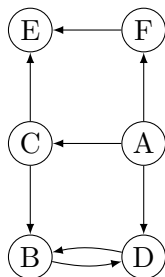
(A,D) (A,E) (B,C) (E,B) (E,F)

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

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

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

**Opgave 165 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

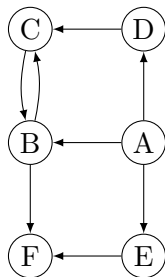
(A,C) (A,F) (B,D) (C,B) (C,E)

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

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

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

**Opgave 166 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

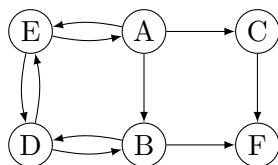
(A,D) (A,E) (B,F) (C,B) (D,C)

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

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

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

**Opgave 167 (Lovlige bredde først træer, 4 %)**



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

Ja Nej

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

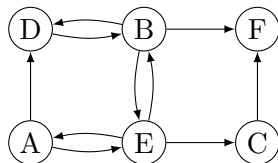
(A,B) (A,C) (A,E) (C,F) (E,D)

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

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

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

**Opgave 168 (Lovlige bredde først træer, 4 %)**

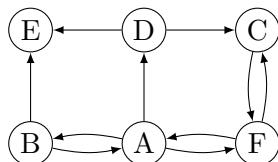


Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

Ja Nej

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

**Opgave 169 (Lovlige bredde først træer, 4 %)**

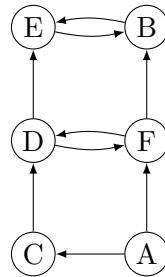


Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

Ja Nej

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

Opgave 170 (Lovlige bredde først træer, 4 %)



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

Ja Nej

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

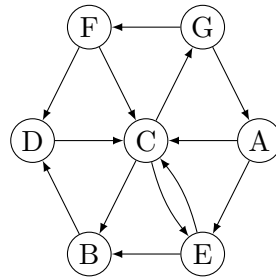
(A,C) (A,F) (C,D) (D,E) (E,B)

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

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

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

Opgave 171 (DFS, 4 %)



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**.

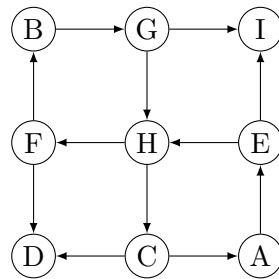
ACBDEGF    ACEBGDF    ACGFEED    ACEBDGF

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

Tree edge    Back edge    Cross edge    Forward edge

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

Opgave 172 (DFS, 4 %)



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**.

AEHICFDBG   AEIHCDFBG   AEHCDFBGI   AEHFBGICD

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

Tree edge   Back edge   Cross edge   Forward edge

(F, D)

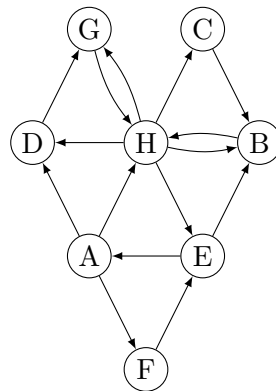
(E, I)

(C, D)

(G, H)



Opgave 173 (DFS, 4 %)



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**.

CBEGHFDA    GDBCEHFA    FECBHGDA    BCEHGDFA

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

Tree edge    Back edge    Cross edge    Forward edge

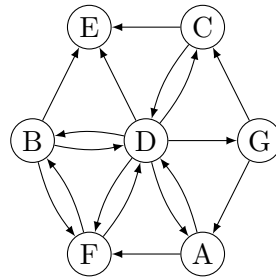
(B, H)

(A, H)

(F, E)

(H, B)

Opgave 174 (DFS, 4 %)



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**.

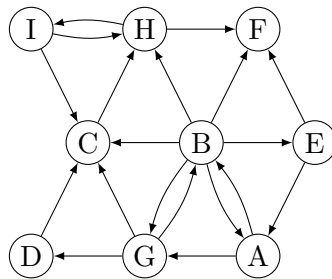
GECBFDA    EBCGDFA    EFBCGDA    GCFEBDA

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

Tree edge    Back edge    Cross edge    Forward edge

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

Opgave 175 (DFS, 4 %)



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**.

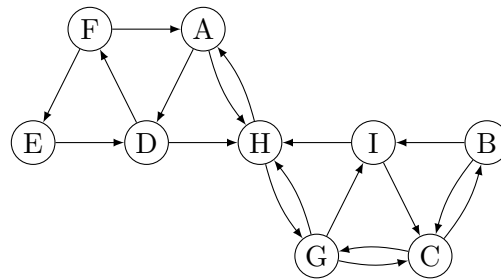
ABCHFIEGD    ABGCHFIDE    ABGCEFHDI    ABGDECHIF

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

Tree edge    Back edge    Cross edge    Forward edge

- (G, B)
- (G, D)
- (A, G)
- (D, C)

Opgave 176 (DFS, 4 %)



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**.

ADHGCBI FE AHGICBDFE ADHFGECIB ADFEHGCBI

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

Tree edge Back edge Cross edge Forward edge

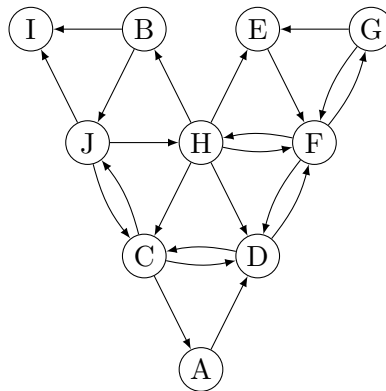
(D, H)

(D, F)

(G, I)

(C, G)

Opgave 177 (DFS, 4 %)



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**.

ADCJHEFGBI    ADCFJGHIEB    ADCJIHBFGE    ADCJHBIEFG

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

Tree edge    Back edge    Cross edge    Forward edge

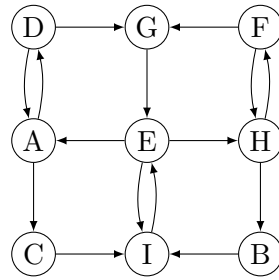
(C, J)

(G, F)

(H, D)

(H, F)

Opgave 178 (DFS, 4 %)



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**.

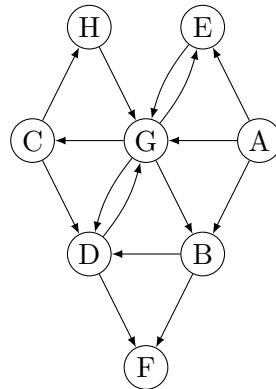
ADCIEHFGB   ACIEHFGBD   ACIEHBFGB   ACDIGEHB

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

Tree edge   Back edge   Cross edge   Forward edge

- (A, D)
- (D, G)
- (A, C)
- (E, I)

Opgave 179 (DFS, 4 %)



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**.

ABEGDFCH    ABDGECHF    AGCDFHEB    ABDFGCHE

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

Tree edge    Back edge    Cross edge    Forward edge

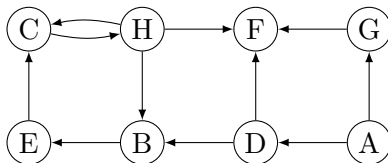
(B, D)

(B, F)

(C, D)

(G, E)

**Opgave 180 (DFS, 4 %)**



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**.

ADGBFECH    ADBECHFG    AGDBECHF    AGFDBECH

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

Tree edge    Back edge    Cross edge    Forward edge

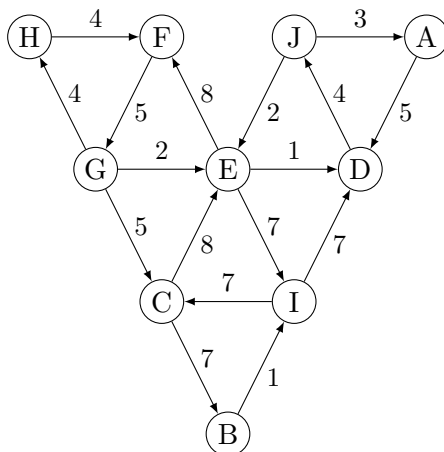
(B, E)

(H, B)

(G, F)

(D, F)

**Opgave 181 (Dijkstras algoritme, 4 %)**

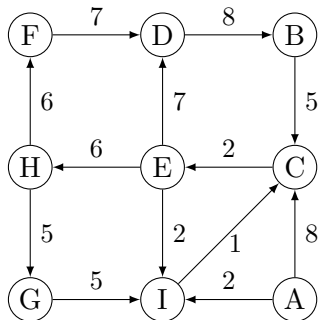


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

ADJEIFGCHB    ADJEICBFGH    ADJEFGCBIH    ADJEFIGCHB



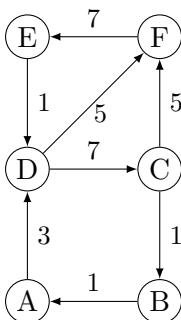
**Opgave 182 (Dijkstras algoritme, 4 %)**



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

ACIEDHBF G ACEDBHF GI AICEHDGFB AICEHGFDB

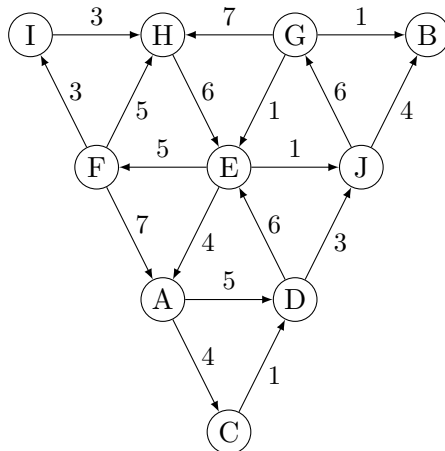
**Opgave 183 (Dijkstras algoritme, 4 %)**



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

ADCFBE ADFECB ADFCBE ADCBFE

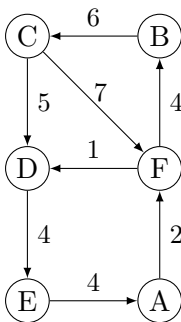
Opgave 184 (Dijkstras algoritme, 4%)



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

ACDJE BGFH ACDFE JIHBG ACDEJ FDBGHI ACDEF HJIJBG

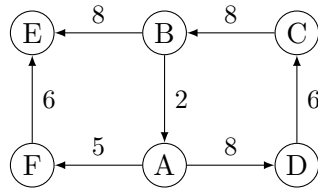
Opgave 185 (Dijkstras algoritme, 4%)



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

AFBDCE AFD BEC AFBCDE AFDEBC

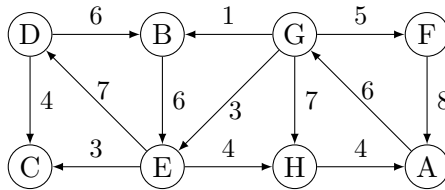
**Opgave 186 (Dijkstras algoritme, 4 %)**



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

ADCBEF    ADFCEB    AFDECB    AFDCBE

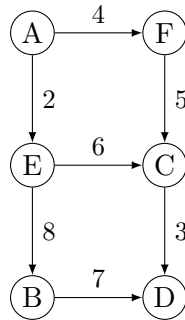
**Opgave 187 (Dijkstras algoritme, 4 %)**



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

AGBEFHCD    AGBECHDF    AGBEFCHD    AGBECDHF

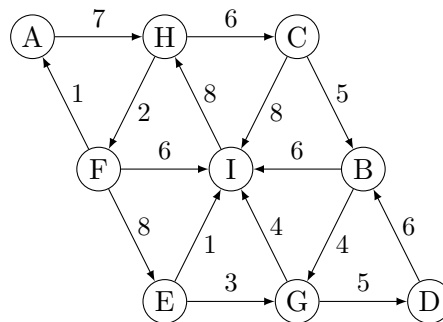
**Opgave 188 (Dijkstras algoritme, 4 %)**



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

- AEFCBD    AEFBCD    AEBDCF    AEFADB

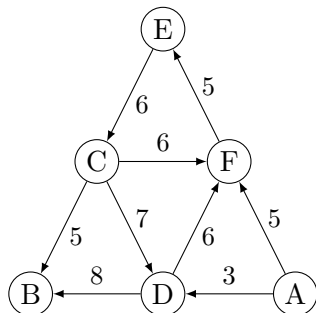
**Opgave 189 (Dijkstras algoritme, 4 %)**



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

- AHCFBIEGD    AHFCIEGDB    AHCBGDIFE    AHFCIEBGD

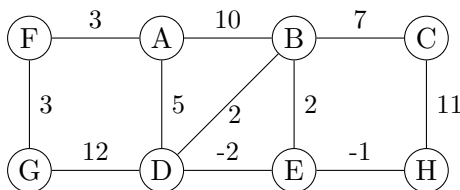
**Opgave 190 (Dijkstras algoritme, 4 %)**



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

- ADBFEC    ADFBEC    ADFEBC    ADFECB

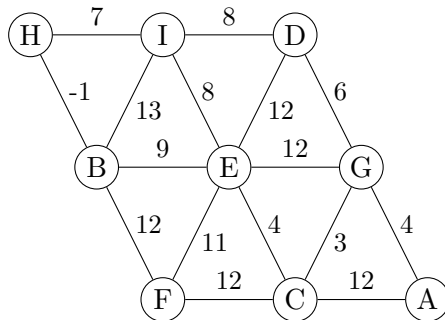
**Opgave 191 (Prims algoritme, 4 %)**



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).

- AFDEHGBC    AFDEHBGC    AFGDEHCB    AFGDEHBC

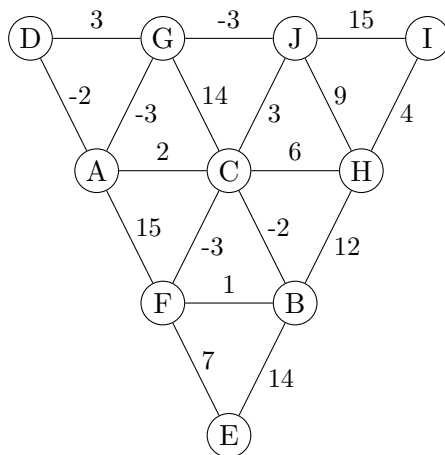
**Opgave 192 (Prims algoritme, 4 %)**



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).

AGCDEIFHB    AGCEIHBFD    AGCEDIHBF    AGCDEIFBH

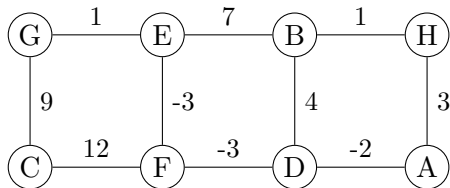
**Opgave 193 (Prims algoritme, 4 %)**



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).

AGJCFBDEHI    AGJDCFBEHI    AGJCFBHIED    AGJDCFBEHI

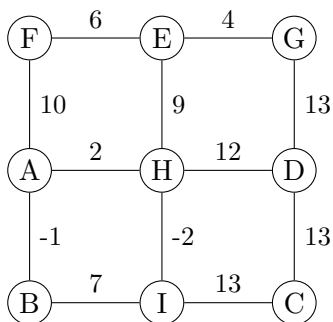
Opgave 194 (Prims algoritme, 4 %)



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).

ADFEHGBC    ADFEGBHC    ADFEGCBH    ADFEGCHB

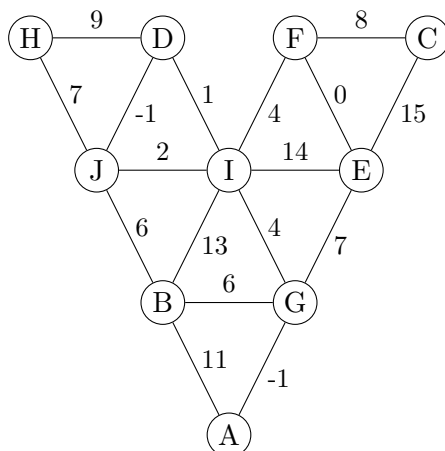
Opgave 195 (Prims algoritme, 4 %)



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).

ABHIFECDG    ABHIEGFDC    ABHIEDGFC    ABIHEGDCF

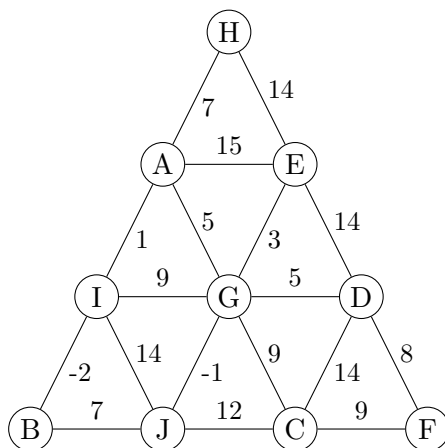
**Opgave 196 (Prims algoritme, 4 %)**



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).

AGIDJBF EHC    AGIDJBHFEC    AGIDJBEFHC    AGIDJFEBHC

**Opgave 197 (Prims algoritme, 4 %)**

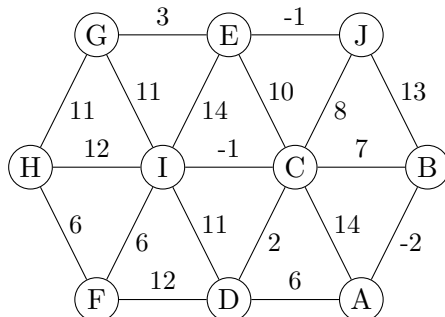


Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

AIBGJHEDFC    AIBJGEDFCH    AIBGJHEDCF    AIBGJEDHFC



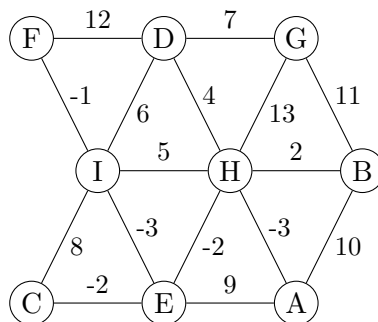
Opgave 198 (Prims algoritme, 4 %)



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).

- ABDCIFJEGH    ABDCIFHJEG    ABCIFHGEJD    ABCIDFJEGH

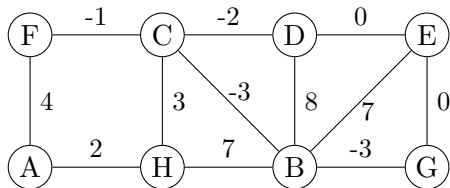
Opgave 199 (Prims algoritme, 4 %)



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).

- AHEIFCDBG    AHEIFDGBC    AHEICFBDG    AHEIFCBDG

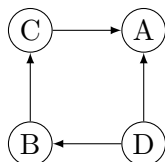
Opgave 200 (Prims algoritme, 4 %)



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).

AHFCBGED    AHCBGDEF    AHCBGEDF    AHCBGDFE

Opgave 201 (Topologisk sortering, 4 %)



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

Ja    Nej

DBCA

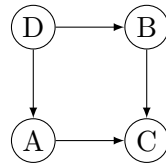
DCBA

CBDA

BDCA

DACB

**Opgave 202 (Topologisk sortering, 4 %)**



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

Ja Nej

DBAC

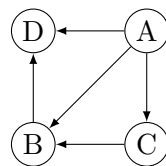
DCAB

DABC

CBAD

ADBC

**Opgave 203 (Topologisk sortering, 4 %)**



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

Ja Nej

ABCD

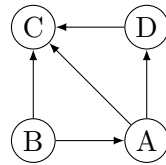
ACBD

BCAD

ADBC

DCBA

**Opgave 204 (Topologisk sortering, 4 %)**



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

Ja Nej

B A D C

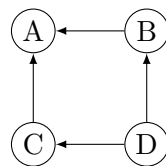
B A C D

C A D B

B C D A

D A B C

**Opgave 205 (Topologisk sortering, 4 %)**



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

Ja Nej

D C A B

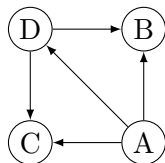
D B C A

A C B D

D C B A

C B D A

Opgave 206 (Topologisk sortering, 4 %)



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

Ja Nej

A D C B

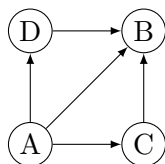
A D B C

C D A B

B D C A

D A B C

Opgave 207 (Topologisk sortering, 4 %)



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

Ja Nej

A C D B

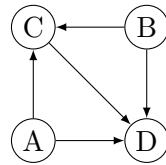
A D B C

C A D B

A D C B

D C A B

**Opgave 208 (Topologisk sortering, 4 %)**



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

Ja Nej

B A D C

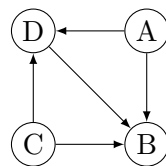
B A C D

A B C D

A D C B

A C B D

**Opgave 209 (Topologisk sortering, 4 %)**



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

Ja Nej

A C D B

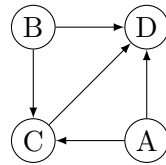
D A C B

C A B D

B A D C

C A D B

**Opgave 210 (Topologisk sortering, 4 %)**



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

Ja Nej

BDCA

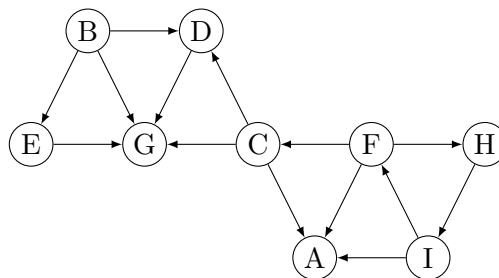
CBAD

BACD

ABCD

DBCA

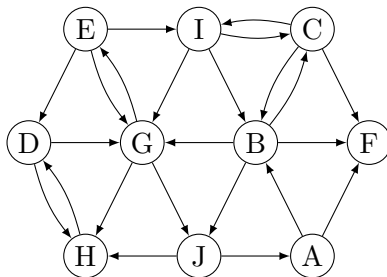
**Opgave 211 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9

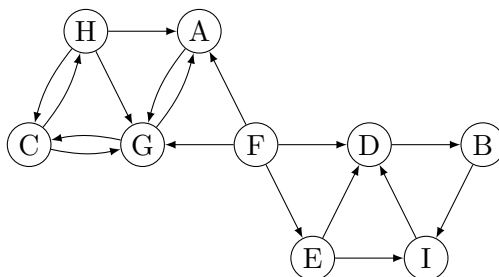
**Opgave 212 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9 10

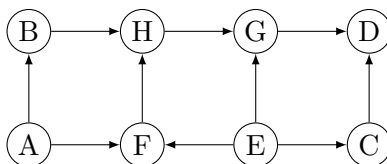
**Opgave 213 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9

**Opgave 214 (Stærke sammenhængskomponenter, 4 %)**

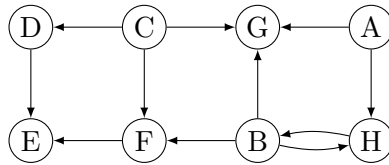


Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8



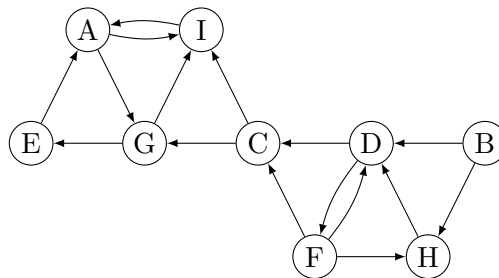
**Opgave 215 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8

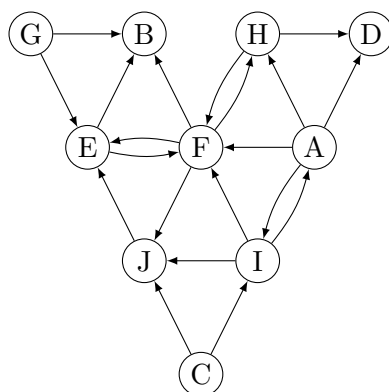
**Opgave 216 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9

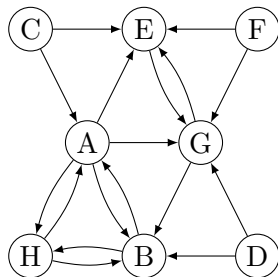
**Opgave 217 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9 10

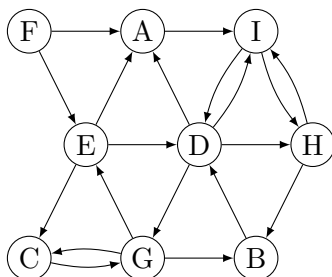
**Opgave 218 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8

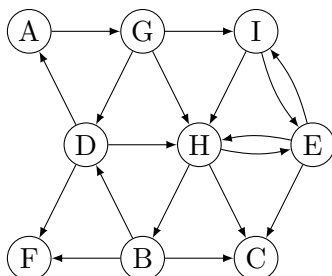
**Opgave 219 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9

**Opgave 220 (Stærke sammenhængskomponenter, 4 %)**



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- 1 2 3 4 5 6 7 8 9

Opgave 221 (Alle løkke opgaver, 0 %)

```

Algoritme loop1(n)   Algoritme loop2(n)   Algoritme loop3(n)
i = 1                 i = 1                 i = 1
while i ≤ n         while i ≤ n         while i ≤ n
    j = 1             j = 1                 j = i
    while j ≤ i       while j ≤ n         while j ≤ n
        j = 2 * j     j = 2 * j             j = 2 * j
    i = 2 * i         i = 2 * i             i = 2 * i

Algoritme loop4(n) Algoritme loop5(n) Algoritme loop6(n)
i = n                 s = 1                 s = 1
while i > 0         for i = 1 to n     for i = 1 to n
    j = i             j = 1                 j = s
    while j > 0       while j ≤ s         while j > 0
        j = ⌊j/2⌋     j = j + 1             s = s + 1
    i = ⌊i/2⌋         s = 2 * s             j = j - 1

Algoritme loop7(n) Algoritme loop8(n) Algoritme loop9(n)
i = 1                 i = 1                 i = 1
p = 1                 j = n                 j = n
while p ≤ n         while i ≤ j         while i ≤ j
    i = i + 1         i = 4 * i             i = i * 2
    p = p * i         j = 2 * j             j = ⌊j/2⌋
    
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

- loop1
- loop2
- loop3
- loop4
- loop5
- loop6
- loop7
- loop8
- loop9

<p><b>Algoritme loop1(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> * <i>i</i> ≤ <i>n</i>              <i>i</i> = <i>i</i> + <i>i</i></p>	<p><b>Algoritme loop2(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> ≤ <i>n</i>              <i>i</i> = 2 * <i>i</i></p>	<p><b>Algoritme loop3(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> ≤ <i>n</i>              <i>i</i> = 3 * <i>i</i></p>
<p><b>Algoritme loop4(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> ≤ <i>n</i>              <i>i</i> = <i>i</i> + <i>i</i></p>	<p><b>Algoritme loop5(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> ≤ <i>n</i> * <i>n</i>              <i>i</i> = 2 * <i>i</i></p>	<p><b>Algoritme loop6(<i>n</i>)</b>  <i>i</i> = 1  <b>while</b> <i>i</i> ≤ <i>n</i> * <i>n</i>              <i>i</i> = 3 * <i>i</i></p>
<p><b>Algoritme loop7(<i>n</i>)</b>  <i>i</i> = <i>n</i>  <b>while</b> <i>i</i> &gt; 0              <b>if</b> <i>i</i> ulige <b>then</b>                  <i>i</i> = <i>i</i> - 1              <b>else</b>                  <i>i</i> = <i>i</i> / 2</p>	<p><b>Algoritme loop8(<i>n</i>)</b>  <i>s</i> = <i>n</i>  <b>while</b> <i>s</i> &gt; 0              <i>s</i> = ⌊<i>s</i> / 2⌋</p>	<p><b>Algoritme loop9(<i>n</i>)</b>  <i>i</i> = 2  <b>while</b> <i>i</i> ≤ <i>n</i>              <i>i</i> = <i>i</i> * <i>i</i></p>

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

Θ(*n* log *n*)   Θ(√*n*)   Θ(log *n*)   Θ(*n*)   Θ(*n*<sup>2</sup>)   Θ(*n*<sup>3</sup>)   Θ(log log *n*)   Θ(2<sup>*n*</sup>)

loop1

loop2

loop3

loop4

loop5

loop6

loop7

loop8

loop9

<p><b>Algoritme loop1(<math>n</math>)</b>  <math>s = 2</math>  <b>while</b> <math>s \leq n</math>              <math>s = s * s</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = 0</math>  <math>s = 0</math>  <math>q = 0</math>  <b>while</b> <math>q \leq n</math>              <math>i = i + 1</math>              <math>s = s + i</math>              <math>q = q + s</math></p>	<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 0</math>  <math>s = 0</math>  <b>while</b> <math>s \leq n</math>              <math>i = i + 1</math>              <math>s = s + i</math></p>
<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>j = 1</math>  <math>s = 0</math>  <b>while</b> <math>s \leq n</math>              <b>while</b> <math>j \leq s</math>                  <math>j = 2 * j</math>              <math>s = s + i</math>              <math>i = i + 1</math></p>	<p><b>Algoritme loop5(<math>n</math>)</b>  <math>j = n</math>  <math>i = 1</math>  <b>while</b> <math>j \geq 0</math>              <math>j = j - i</math>              <math>i = i + 1</math></p>	<p><b>Algoritme loop6(<math>n</math>)</b>  <math>s = 0</math>  <math>i = 1</math>  <b>while</b> <math>s \leq n</math>              <math>s = s + i</math>              <math>i = i + 1</math></p>
<p><b>Algoritme loop7(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>              <math>j = 1</math>              <math>k = 1</math>              <b>while</b> <math>k \leq n</math>                  <math>j = j + 1</math>                  <math>k = k + j</math>              <math>i = 2 * i</math></p>	<p><b>Algoritme loop8(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = i</math>              <b>while</b> <math>j \leq n</math>                  <math>j = 2 * j</math></p>	<p><b>Algoritme loop9(<math>n</math>)</b>  <math>i = 0</math>  <math>j = n</math>  <b>while</b> <math>i \leq j</math>              <math>i = i + 1</math>              <math>j = j - 1</math></p>

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

loop5

loop6

loop7

loop8

loop9

<p><b>Algoritme loop1(<math>n</math>)</b>  <math>i = 1</math>  <math>j = 0</math>  <b>while</b> <math>i \leq n</math>          <math>i = i + i</math>          <b>while</b> <math>j &lt; i</math>              <math>j = j + 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = 1</math>  <math>j = 1</math>  <b>while</b> <math>i \leq n</math>          <b>while</b> <math>j \leq i</math>              <math>j = j + 1</math>          <math>i = 2 * i</math></p>	<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 1</math>  <math>s = 0</math>  <b>while</b> <math>s \leq n</math>          <math>j = 1</math>          <b>while</b> <math>j \leq i</math>              <math>j = j + 1</math>          <math>s = s + i</math>          <math>i = i + 1</math></p>
<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>s = 1</math>  <b>while</b> <math>s \leq n * n</math>          <math>i = i + 1</math>          <math>s = s + i</math></p>	<p><b>Algoritme loop5(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <math>j = 0</math>          <b>while</b> <math>j \leq i</math>              <math>j = j + 1</math>          <math>i = 2 * i</math></p>	<p><b>Algoritme loop6(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <math>j = 0</math>          <b>while</b> <math>j \leq n</math>              <math>j = j + i</math>          <math>i = 2 * i</math></p>
<p><b>Algoritme loop7(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <math>j = 1</math>          <b>while</b> <math>j \leq i</math>              <math>j = j + 1</math>          <math>i = 2 * i</math></p>	<p><b>Algoritme loop8(<math>n</math>)</b>  <math>i = n</math>  <b>while</b> <math>i &gt; 0</math>          <math>i = i - 1</math></p>	<p><b>Algoritme loop9(<math>n</math>)</b>  <math>i = n</math>  <b>while</b> <math>i \geq 1</math>          <math>j = i</math>          <b>while</b> <math>j \leq n</math>              <math>j = 2 * j</math>          <math>i = i - 1</math></p>

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

- loop1
- loop2
- loop3
- loop4
- loop5
- loop6
- loop7
- loop8
- loop9

<p><b>Algoritme loop1(<math>n</math>)</b>  <math>s = 0</math>  <math>i = 1</math>  <b>while</b> <math>i * i \leq n</math>          <b>for</b> <math>j = 1</math> <b>to</b> <math>i</math>              <math>s = s + 1</math>          <math>i = i + 1</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>s = 0</math>  <math>i = n</math>  <b>while</b> <math>i &gt; 1</math>          <b>for</b> <math>j = 1</math> <b>to</b> <math>i</math>              <math>s = s + 1</math>          <math>i = \lfloor i/2 \rfloor</math></p>	<p><b>Algoritme loop3(<math>n</math>)</b>  <math>s = 1</math>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>          <math>s = s + 1</math></p>
<p><b>Algoritme loop4(<math>n</math>)</b>  <math>s = 1</math>  <b>for</b> <math>i = n</math> <b>to</b> <math>1</math> <b>step</b> <math>-1</math>          <math>s = s + 1</math></p>	<p><b>Algoritme loop5(<math>n</math>)</b>  <math>s = 1</math>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>          <b>for</b> <math>j = 1</math> <b>to</b> <math>i</math>              <math>s = s + 1</math>          <math>i = 2 * i</math></p>	<p><b>Algoritme loop6(<math>n</math>)</b>  <math>s = 1</math>  <b>while</b> <math>s \leq n</math>          <math>s = s + 1</math></p>
<p><b>Algoritme loop7(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>          <math>j = 0</math>          <b>while</b> <math>j \leq n</math>              <math>j = j + i</math></p>	<p><b>Algoritme loop8(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>          <math>j = 1</math>          <b>while</b> <math>j \leq i</math>              <math>j = 2 * j</math></p>	<p><b>Algoritme loop9(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>          <math>j = 1</math>          <b>while</b> <math>j \leq n</math>              <math>j = 2 * j</math></p>

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af  $n$  i  $\Theta$ -notation.

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

- loop1
- loop2
- loop3
- loop4
- loop5
- loop6
- loop7
- loop8
- loop9

<p><b>Algoritme loop1(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = i</math>              <b>while</b> <math>j &gt; 1</math>                  <math>j = \lfloor j/2 \rfloor</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <math>i = 0</math>              <b>while</b> <math>i \leq n</math>                  <math>j = i</math>                  <b>while</b> <math>j &gt; 0</math>                      <math>j = \lfloor j/2 \rfloor</math>                  <math>i = i + 1</math></p>	<p><b>Algoritme loop3(<math>n</math>)</b>  <math>i = 1</math>  <math>j = 1</math>  <math>s = 0</math>              <b>while</b> <math>i \leq n</math>                  <b>if</b> <math>i = j</math> <b>then</b>                      <b>for</b> <math>k = 1</math> <b>to</b> <math>n</math>                          <math>s = s + 1</math>                          <math>j = 2 * j</math>                  <math>i = i + 1</math></p>
<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 1</math>  <math>s = 0</math>              <b>while</b> <math>i \leq n</math>                  <b>for</b> <math>j = i</math> <b>to</b> <math>n</math>                      <math>s = s + 1</math>                  <math>i = i + i</math></p>	<p><b>Algoritme loop5(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n</math>                  <math>j = 0</math>                  <b>while</b> <math>j \leq n</math>                      <math>j = j + 1</math>                  <math>i = 2 * i</math></p>	<p><b>Algoritme loop6(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n</math>                  <math>j = 1</math>                  <b>while</b> <math>j \leq i</math>                      <math>j = 2 * j</math>                  <math>i = i + 1</math></p>
<p><b>Algoritme loop7(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n</math>                  <math>j = i</math>                  <b>while</b> <math>j \leq n</math>                      <math>j = j + 1</math>                  <math>i = 2 * i</math></p>	<p><b>Algoritme loop8(<math>n</math>)</b>  <math>i = 1</math>              <b>while</b> <math>i \leq n</math>                  <math>j = n</math>                  <b>while</b> <math>j &gt; 1</math>                      <math>j = j - 1</math>                  <math>i = 2 * i</math></p>	<p><b>Algoritme loop9(<math>n</math>)</b>  <math>s = 0</math>  <math>i = n</math>              <b>while</b> <math>i &gt; 1</math>                  <b>for</b> <math>j = 1</math> <b>to</b> <math>n</math>                      <math>s = s + 1</math>                  <math>i = \lfloor i/2 \rfloor</math></p>

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

loop5

loop6

loop7

loop8

loop9



<p><b>Algoritme loop1(<math>n</math>)</b>  <b>for</b> <math>i = 0</math> <b>to</b> <math>n</math>              <math>j = 0</math>              <math>s = 0</math>              <b>while</b> <math>s \leq i</math>                  <math>j = j + 1</math>                  <math>s = s + j</math></p>	<p><b>Algoritme loop2(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq i</math>                  <math>j = j + 1</math></p>	<p><b>Algoritme loop3(<math>n</math>)</b>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <math>j = i</math>              <b>while</b> <math>j &gt; 0</math>                  <math>j = j - 1</math></p>
<p><b>Algoritme loop4(<math>n</math>)</b>  <math>i = 0</math>  <math>j = 0</math>  <b>while</b> <math>i \leq n</math>              <b>if</b> <math>i &lt; j</math> <b>then</b>                  <math>i = i + 1</math>              <b>else</b>                  <math>j = j + 1</math>              <math>i = 0</math></p>	<p><b>Algoritme loop5(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq i</math>                  <math>j = j + 1</math>              <math>i = i + 1</math></p>	<p><b>Algoritme loop6(<math>n</math>)</b>  <math>i = 1</math>  <b>while</b> <math>i \leq n</math>              <math>j = 1</math>              <b>while</b> <math>j \leq n</math>                  <math>j = j + 1</math>              <math>i = i + 1</math></p>
<p><b>Algoritme loop7(<math>n</math>)</b>  <math>s = 0</math>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <b>for</b> <math>j = 1</math> <b>to</b> <math>n</math>                  <b>if</b> <math>i = j</math> <b>then</b>                      <b>for</b> <math>k = 1</math> <b>to</b> <math>n</math>                          <math>s = s + 1</math></p>	<p><b>Algoritme loop8(<math>n</math>)</b>  <math>s = 0</math>  <math>i = n</math>  <b>while</b> <math>i &gt; 0</math>              <b>for</b> <math>j = 1</math> <b>to</b> <math>i</math>                  <math>s = s + 1</math>              <math>i = i - 1</math></p>	<p><b>Algoritme loop9(<math>n</math>)</b>  <math>s = 1</math>  <b>for</b> <math>i = 1</math> <b>to</b> <math>n</math>              <b>for</b> <math>j = 1</math> <b>to</b> <math>n</math>                  <math>s = s + 1</math></p>

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

- loop1
- loop2
- loop3
- loop4
- loop5
- loop6
- loop7
- loop8
- loop9

**Algoritme loop1( $n$ )**

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

**Algoritme loop2( $n$ )**

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

**Algoritme loop3( $n$ )**

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

**Algoritme loop4( $n$ )**

```
for i = 1 to n
  for j = 1 to i
    k = 1
    while k ≤ i + j
      k = 2 * k
```

**Algoritme loop5( $n$ )**

```
s = 0
for i = 1 to n
  for j = 1 to i * i
    s = s + 1
```

**Algoritme loop6( $n$ )**

```
s = 0
for i = 1 to n
  for j = 1 to n
    for k = 1 to n
      s = s + 1
```

**Algoritme loop7( $n$ )**

```
s = 0
for i = 1 to n
  for j = i to n
    for k = i to j
      s = s + 1
```

**Algoritme loop8( $n$ )**

```
s = 0
j = 0
for i = 1 to n
  j = j + i
  for k = 1 to j
    s = s + 1
```

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af  $n$  i  $\Theta$ -notation.

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

loop1

loop2

loop3

loop4

loop5

loop6

loop7

loop8