

Opgave 1 (4%)

|                                     | Ja                                  | Nej                                 |
|-------------------------------------|-------------------------------------|-------------------------------------|
| $4n^8$ er $O(8 \cdot 4^n)$ ?        | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $4^{n/2}$ er $O(2^n)$ ?             | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $(\log n)/n^3$ er $O(1)$ ?          | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $n^2 + 3^n$ er $O(2^n)$ ?           | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\sqrt{n} \cdot \log n$ er $O(n)$ ? | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |

Opgave 2 (4%)

Opskriv følgende funktioner efter stigende orden med hensyn til  $O$ -notationen:

$(\sqrt{n})^{\log n}$   
 $\sqrt{n}/\log n$   
 $1/(n^2)$   
 $n^7$   
 $42n$

Svar: \_\_\_\_\_  $1/(n^2)$      $\sqrt{n}/\log n$      $42n$      $n^7$      $(\sqrt{n})^{\log n}$

Opgave 3 (4%)

Angiv for hver af nedenstående algoritmer udførelstiden som funktion af  $n$  i  $O$ -notation.

**Algoritme Loop1( $n$ )**  
 $x \leftarrow 0$   
**for**  $i \leftarrow 1$  **to**  $n$  **do**  
  **for**  $j \leftarrow 1$  **to**  $n$  **do**  
    **for**  $k \leftarrow 1$  **to**  $n$  **do**  
       $x \leftarrow x + 1$

**Algoritme Loop2( $n$ )**  
 $x \leftarrow 0$   
**for**  $i \leftarrow 1$  **to**  $n$  **do**  
  **for**  $j \leftarrow 1$  **to**  $i$  **do**  
    **for**  $k \leftarrow j$  **to**  $n$  **do**  
       $x \leftarrow x + 1$

**Algoritme Loop3( $n$ )**  
 $i \leftarrow 1$   
**while**  $i \leq n$  **do**  
   $j \leftarrow 1$   
  **while**  $j \leq n$  **do**  
     $k \leftarrow 1$   
    **while**  $k \leq n$  **do**  
       $k \leftarrow k + 1$   
     $j \leftarrow j * 2$   
   $i \leftarrow i * 2$

Svar Loop1: \_\_\_\_\_  $O(n^3)$

Svar Loop2: \_\_\_\_\_  $O(n^3)$

Svar Loop3: \_\_\_\_\_  $O(n(\log n)^2)$

**Opgave 4 (4%)**

Angiv for hver af nedenstående algoritmer udførelstiden som funktion af  $n$  i  $O$ -notation.

**Algoritme Loop1( $n$ )**

```

i ← n
j ← 1
while i ≥ 1 do
    k ← n
    l ← 1
    while k ≥ 1 do
        l ← l + 1
        k ← k - l
    j ← j + 1
    i ← i - j
    
```

**Algoritme Loop2( $n$ )**

```

i ← 1
while i ≤ n do
    k ← 1
    l ← 1
    while k ≤ n do
        l ← l + 1
        k ← k + l
    i ← i * 2
    
```

**Algoritme Loop2( $n$ )**

```

i ← 1
j ← 1
while i ≤ n do
    i ← i * 2
    while j < i do
        j ← j + 1
    
```

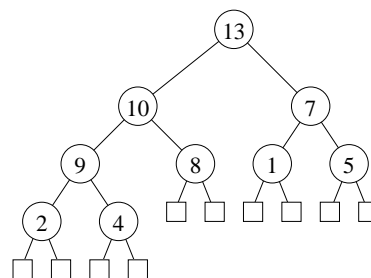
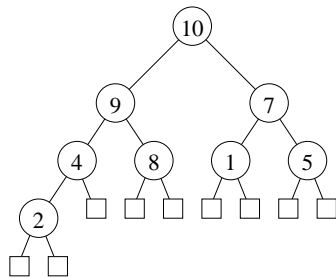
Svar Loop1: \_\_\_\_\_  $O(n)$  \_\_\_\_\_

Svar Loop2: \_\_\_\_\_  $O(\log n \sqrt{n})$  \_\_\_\_\_

Svar Loop3: \_\_\_\_\_  $O(n)$  \_\_\_\_\_

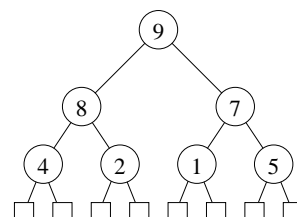
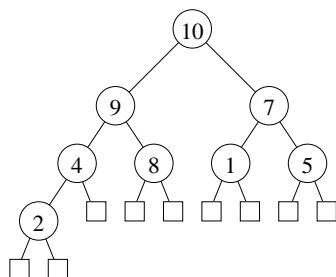
**Opgave 5 (4%)**

Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 13.



Svar: \_\_\_\_\_

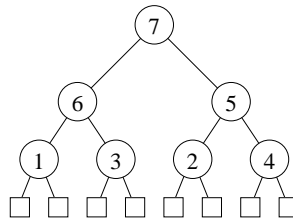
Tegn hvordan nedenstående binære max-heap ser ud efter en heap-extract-max operation.



Svar: \_\_\_\_\_

**Opgave 6 (4%)**

Tegn den binære max-heap efter indsættelse af elementerne 1, 7, 2, 6, 3, 4, 5 i den givne rækkefølge, startende med den tomme heap.



Svar: \_\_\_\_\_

**Opgave 7 (4%)**

Angiv hvordan nedenstående array ser ud efter anvendelsen af build-max-heap for arrayet.

|   |   |   |   |   |   |    |    |   |    |    |    |    |
|---|---|---|---|---|---|----|----|---|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7  | 8  | 9 | 10 | 11 | 12 | 13 |
| 7 | 2 | 3 | 9 | 4 | 1 | 10 | 11 | 5 | 8  | 6  | 13 | 12 |

|    |    |    |   |   |   |    |   |   |    |    |    |    |
|----|----|----|---|---|---|----|---|---|----|----|----|----|
| 1  | 2  | 3  | 4 | 5 | 6 | 7  | 8 | 9 | 10 | 11 | 12 | 13 |
| 13 | 11 | 12 | 9 | 8 | 7 | 10 | 2 | 5 | 4  | 6  | 1  | 3  |

Svar: \_\_\_\_\_

**Opgave 8 (4%)**

Betragt en binær max-heap med  $n$  forskellige elementer. Angiv hvilke af følgende udsagn er sande.

- |  | Ja                                  | Nej                                 |
|--|-------------------------------------|-------------------------------------|
| Det mindste element kan findes i tid $O(1)$                | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Det største element kan findes i tid $O(\log n)$           | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| Højden af en binær max-heap er $\lceil \log n \rceil$      | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Det næststørste og tredje største element er børn af roden | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Man kan søge efter et element $x$ i tid $O(\log n)$        | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

**Opgave 9 (4%)**

Hvad er udførelstiden for indsættelse af følgende sekvens af  $n$  elementer i nedenstående datastrukturer? Det antages at elementerne indsættes enkeltvis i den givne rækkefølge.

a)  $1, 2, 3, \dots, n - 1, n$

|                        |                                       |
|------------------------|---------------------------------------|
| a) Ubalanceret søgetræ | Svar: <u><math>O(n^2)</math></u>      |
| b) Rød-sort søgetræ    | Svar: <u><math>O(n \log n)</math></u> |
| c) Heap                | Svar: <u><math>O(n \log n)</math></u> |
| d) Stak                | Svar: <u><math>O(n)</math></u>        |

**Opgave 10 (4%)**

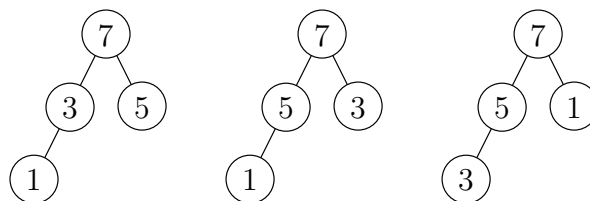
Betragt radix-sort anvendt på nedenstående liste af tal ( $d = 4, k = 10$ ). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter de to mindst betydende cifre.

4569 1778 8327 4353 9913 4227 4469

Svar: 9913 8327 4227 4353 4569 4469 1778

**Opgave 11 (4%)**

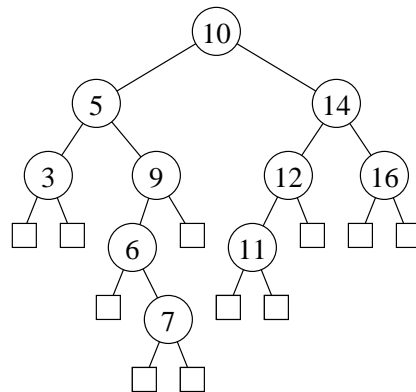
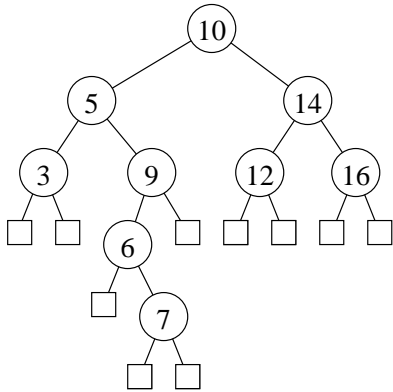
Angiv alle mulige binære max-heaps for mængden  $\{1, 3, 5, 7\}$ .



Svar: \_\_\_\_\_

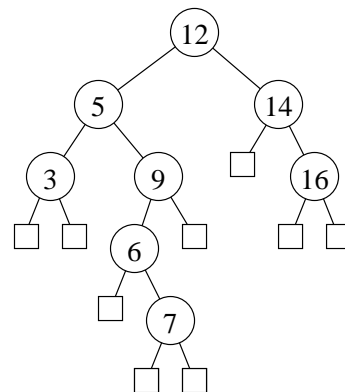
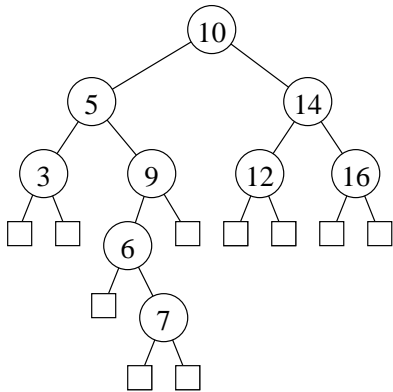
**Opgave 12 (4%)**

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 11.



Svar: \_\_\_\_\_

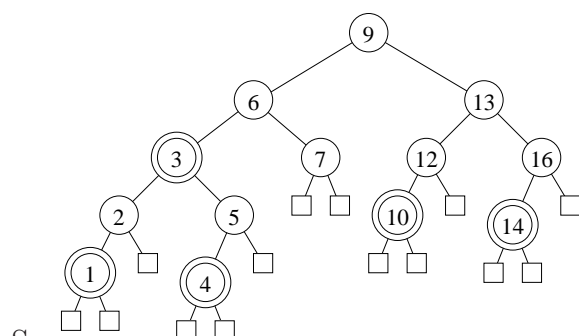
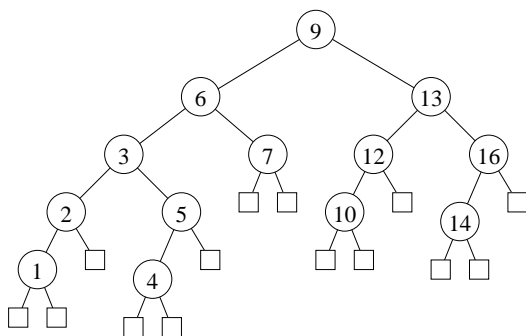
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter slettelse af elementet 10.



Svar: \_\_\_\_\_

**Opgave 13 (4%)**

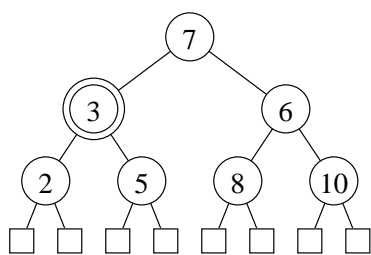
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.



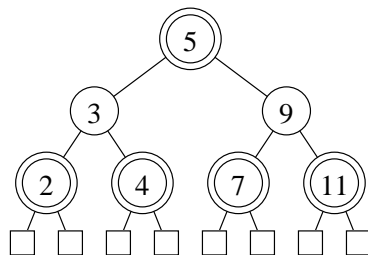
Svar: \_\_\_\_\_

**Opgave 14 (4%)**

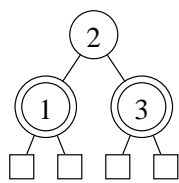
Angiv for hvert af nedenstående træer om det er et lovligt søgetræ, et lovligt rød-sort søgetræ, eller ingen af delene (dobbeltcirkler angiver røde knuder).



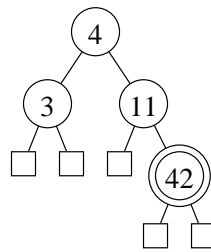
a)



b)



c)

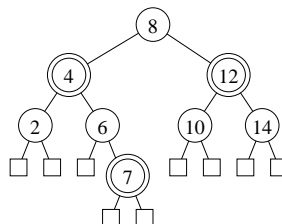
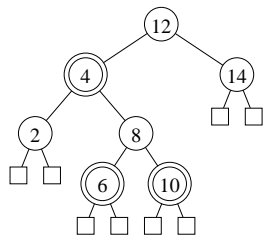


d)

|    | Rød-sort søgetræ                    | Søgetræ, men ikke rød-sort          | Ikke et søgetræ                     |
|----|-------------------------------------|-------------------------------------|-------------------------------------|
| a) | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| b) | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| c) | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |
| d) | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |

**Opgave 15 (4%)**

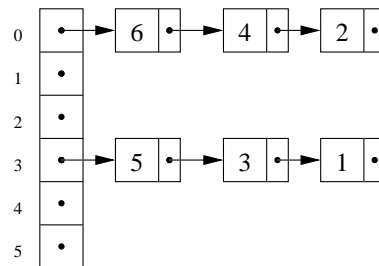
Tegn hvordan nedenstående rød-sort træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 7.



Svar: \_\_\_\_\_

**Opgave 16 (4%)**

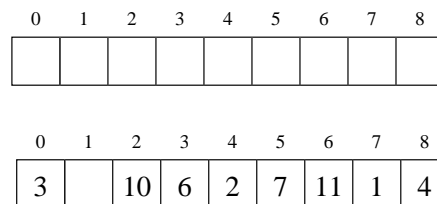
Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hash-funktionen er  $h(k) = k * 3 \text{ mod } 6$  og der indsættes elementerne 1, 2, 3, 4, 5 og 6 i den givne rækkefølge.



Svar: \_\_\_\_\_

**Opgave 17 (4%)**

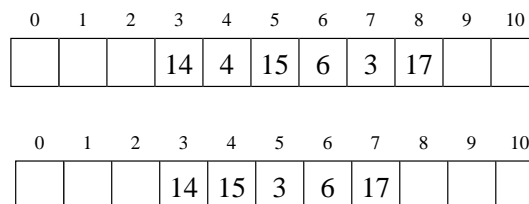
Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 2, 7, 6, 10, 11, 1, 4, og 3 indsættes i den givne rækkefølge, når hashfunktionen er  $h(k) = 2 * k \text{ mod } 9$ .



Svar: \_\_\_\_\_

**Opgave 18 (4%)**

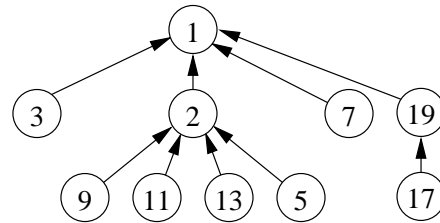
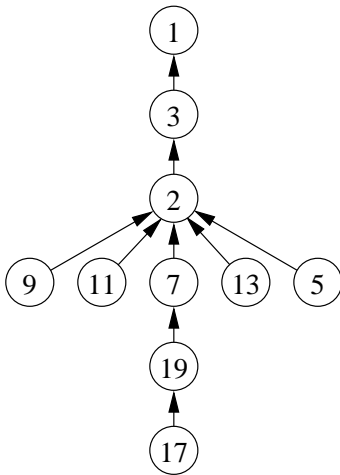
I nedenstående hashtabel er der anvendt *linear probing* med hashfunktionen  $h(k) = k \text{ mod } 11$ . Tegn hvordan hashtabellen kan se ud efter at 4 slettes.



Svar: \_\_\_\_\_

**Opgave 19 (4%)**

Tegn hvordan nedenstående union-find datastruktur ser ud efter FIND(19), når der anvendes stikomprimering.



Svar: \_\_\_\_\_

**Opgave 20 (4%)**

Hvor lang tid tager det at slette det største element fra:

a) et rød-sort søgetræ

Svar:  $O(\log n)$

b) en min-heap

Svar:  $O(n)$

c) en usorteret cyclisk dobbelt kædet liste

Svar:  $O(n)$

d) en sorteret cyclisk dobbelt kædet liste

Svar:  $O(1)$



**Transitionssystem** DownSum  
Konfigurationer:  $\{[i, n, s] \mid i, n, s \geq 0\}$   
 $[i, n, s] \triangleright [i - 1, n, s + 2i - 1] \quad \text{if } i \geq 1$

**Opgave 21** (4%)

For hvert af nedenstående udsagn, angiv om de er en invariant for ovenstående transitionssystem DownSum. Startkonfigurationen antages at være  $[n, n, 0]$  hvor  $n \geq 0$ .

|                 | Ja                                  | Nej                                 |
|-----------------|-------------------------------------|-------------------------------------|
| $i \leq n$      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $s \geq n$      | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $i - s \geq 0$  | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $s = i^2$       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $s = n^2 - i^2$ | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |

**Opgave 22** (4%)

For hver af nedenstående funktioner, angiv om de er en termineringsfunktion for ovenstående transitionssystem DownSum.

|                                | Ja                                  | Nej                                 |
|--------------------------------|-------------------------------------|-------------------------------------|
| $\mu(i, n, s) = i$             | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $\mu(i, n, s) = n - i$         | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\mu(i, n, s) = s$             | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\mu(i, n, s) = n^2 - s$       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $\mu(i, n, s) = n^2 - i^2 - s$ | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

**Algoritme** Loop( $n$ )

Inputbetingelse : heltal  $n \geq 1$

Outputkrav : –

Metode :  $i \leftarrow 0$ ;

$j \leftarrow 0$ ;

$\{I\}$  **while**  $i < n$  **or**  $j < n$  **do**  
    **if**  $i < j$  **do**  $i \leftarrow i + 2$ ;  
    **else**  $j \leftarrow j + 3$

**Opgave 23** (4%)

For hver af nedenstående udsagn, angiv om de er en invariant  $I$  for ovenstående algoritme Loop.

|                           | Ja                                  | Nej                                 |
|---------------------------|-------------------------------------|-------------------------------------|
| $i \leq n$                | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $j \geq i - 1$            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $i + j \leq 2n + 3$       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $\min\{i, j\} \leq n + 1$ | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $i * j \leq n^2$          | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

**Opgave 24** (4%)

For hver af nedenstående funktioner, angiv om de er en termineringsfunktion for ovenstående algoritme Loop.

|                                  | Ja                                  | Nej                                 |
|----------------------------------|-------------------------------------|-------------------------------------|
| $\mu(i, j, n) = n - i$           | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\mu(i, j, n) = 2n + 3 - i - j$  | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| $\mu(i, j, n) = j - i$           | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\mu(i, j, n) = n^2 - i^2 - j^2$ | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| $\mu(i, j, n) = i^2 - j^2$       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

**Opgave 25 (4%)**

Nedenstående algoritme beregner  $n^2$  for et heltal  $n \geq 0$ . For at vise gyldigheden af algoritmen skal  $I_i$  og  $I_s$  være invarianter omkring variablerne  $i$  og  $s$ . Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke). Det antages at  $n$  ikke kan ændres af algoritmen.

```
Algoritme Square( $n$ )  
Inputbetingelse : Heltal  $n \geq 0$   
Outputkrav    :  $s = n^2$   
Metode        :  $s = 0$ ;  
                $i = 1$ ;  
                $\{I_i \wedge I_s\}$  while  $i \leq n$  do  
                    $s \leftarrow s + i$ ;  
                    $i \leftarrow i + 1$ ;  
                $s \leftarrow 2 * s - n$ 
```

Svar  $I_i$ :                      $1 \leq i \leq n + 1$                     

Svar  $I_s$ :                      $s = \sum_{j=0}^{i-1} j = \frac{i^2-i}{2}$                     

For at kunne bevise at algoritmen terminerer, kræves en passende termineringsfunktion. Angiv en termineringsfunktion (bevis for at termineringsfunktionen har de nødvendige egenskaber kræves ikke).

Svar  $\mu$ :                      $n + 1 - i$