

Opgave 4 (4%)

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

Algoritme Loop1(n)

```
for  $i = 1$  to  $n$ 
   $s = 0$ 
  while  $s \leq i$ 
     $s = s + 1$ 
```

Algoritme Loop2(n)

```
for  $i = 1$  to  $n$ 
   $j = 1$ 
   $s = 1$ 
  while  $s \leq i$ 
     $j = j + 1$ 
     $s = s + j$ 
```

Algoritme Loop3(n)

```
 $s = 1$ 
while  $s \leq n$ 
   $s = 2 * s$ 
```

Svar Loop1: _____ $O(n^2)$ _____

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

Svar Loop3: _____ $O(\log n)$ _____

Opgave 5 (4%)

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

Algoritme Loop1(n)

```
 $i = 1$ 
while  $i \leq n$ 
   $j = 0$ 
  while  $j \leq n$ 
     $j = j + i$ 
   $i = 2 * i$ 
```

Algoritme Loop2(n)

```
 $i = 1$ 
 $s = 1$ 
while  $s \leq n * n$ 
   $i = i + 1$ 
   $s = s + i$ 
```

Algoritme Loop3(n)

```
 $i = n$ 
while  $i \geq 1$ 
   $j = i$ 
  while  $j \leq n$ 
     $j = 2 * j$ 
   $i = i - 1$ 
```

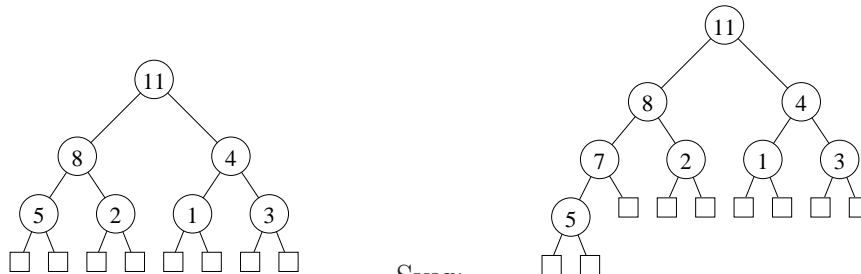
Svar Loop1: _____ $O(n)$ _____

Svar Loop2: _____ $O(n)$ _____

Svar Loop3: _____ $O(n)$ _____

Opgave 6 (4%)

Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 7.



Svar: _____

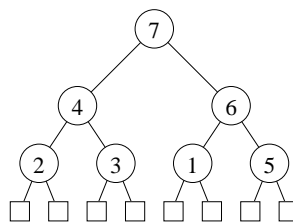
Tegn hvordan nedenstående binære max-heap ser ud efter en HEAP-EXTRACT-MAX operation.



Svar: _____

Opgave 7 (4%)

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



Svar: _____

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

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

Svar: _____

Opgave 9 (4%)

Betragt RADIX-SORT anvendt på nedenstående liste af tal ($d = 3, k = 10$). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter det mindst betydende ciffer.

814 371 991 562 744 241 432

Svar: 371 991 241 562 432 814 744

Opgave 10 (4%)

Angiv resultatet af at anvende PARTITION($A, 8, 19$) på nedenstående array.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	8	16	1	6	2	4	13	17	23	3	18	5	9	11	24	12	14	10	15	22

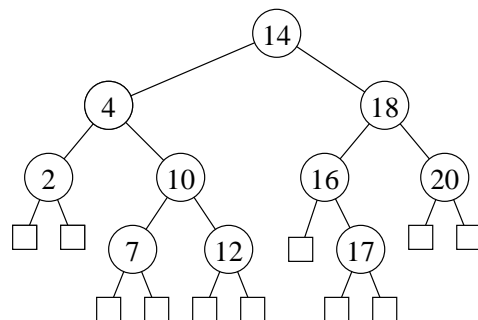
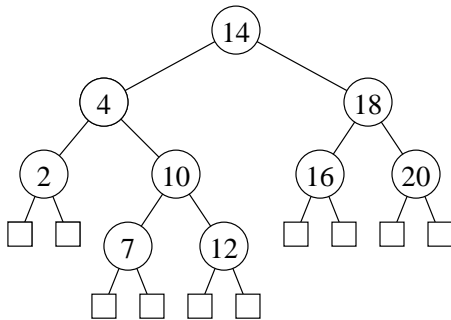


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Svar:	8	16	1	6	2	4	13	3	5	9	11	12	14	10	15	23	17	18	24	22

Svar: _____

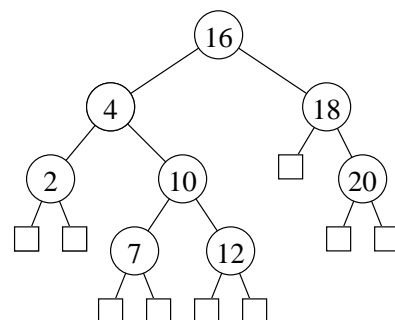
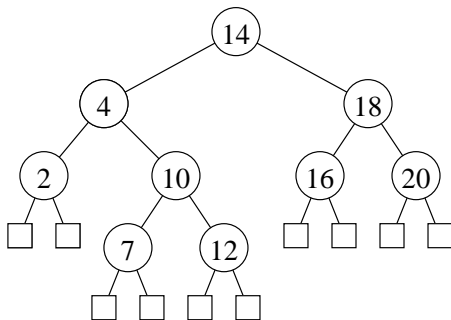
Opgave 11 (4%)

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 17.



Svar: _____

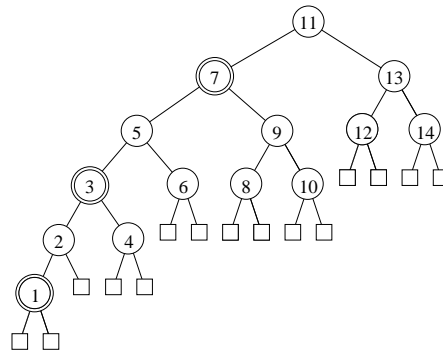
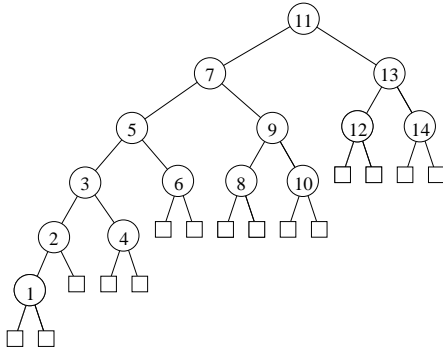
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter slettelse af elementet 14.



Svar: _____

Opgave 12 (4%)

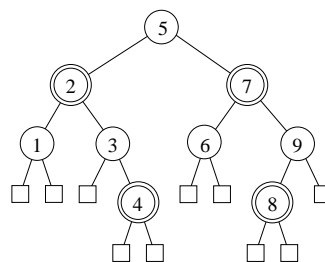
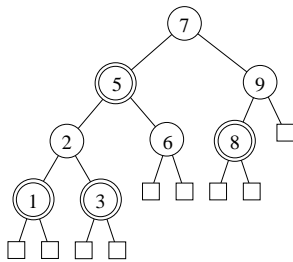
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.



Svar: _____

Opgave 13 (4%)

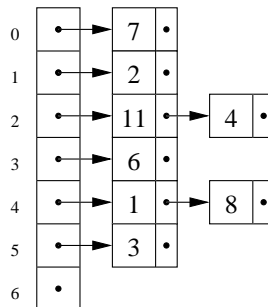
Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 4.



Svar: _____

Opgave 14 (4%)

Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hash-funktionen er $h(k) = 4k \text{ mod } 7$ og der indsættes elementerne 3, 2, 4, 8, 6, 11, 1 og 7 i den givne rækkefølge.



Svar: _____

Opgave 15 (4%)

Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 8, 2, 3, 1, 12, 16, 7 og 5 indsættes i den givne rækkefølge, når hashfunktionen er $h(k) = 2k \bmod 11$.

0	1	2	3	4	5	6	7	8	9	10

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

Svar: _____

Opgave 16 (4%)

Tegn hvordan en hashtabel der anvender *dobbelt hashing* ser ud efter at elementerne 17, 20, 4, 8, 7, 12 og 6 indsættes i den givne rækkefølge, når hashfunktionerne er $h_1(k) = k \bmod 13$ og $h_2(k) = 1 + (2k \bmod 7)$, og hashtabellen har størrelse 13.

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

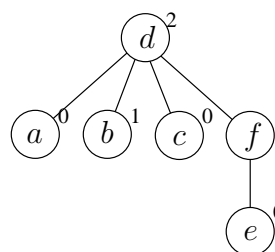
0	1	2	3	4	5	6	7	8	9	10	11	12
				17	6	4	20	8	7			12

Svar: _____

Opgave 17 (4%)

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

- makeset(a)
- makeset(b)
- makeset(c)
- makeset(d)
- makeset(e)
- makeset(f)
- union(a, b)
- union(c, d)
- union(a, c)
- union(e, f)
- union(a, e)



Svar: _____

Opgave 18 (4%)

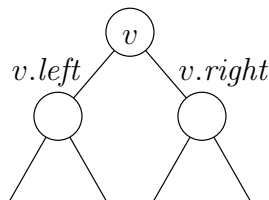
Betragt et søgetræ hvor hver knude v ud over et element $v.e$ gemmer en værdi $v.x$ (søgetræet er ordnet efter $v.e$). For at anvende træet til statistiske formål udvider vi hver knude v med følgende information:

$$v.size = \text{antal knuder i } v\text{'s undertræ } T_v$$

$$v.sum = \sum_{u \in T_v} u.x$$

$$v.sumofsquares = \sum_{u \in T_v} (u.x)^2$$

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



$$\text{Svar } v.size = \frac{v.left.size + v.right.size + 1}{}$$

$$\text{Svar } v.sum = \frac{v.left.sum + v.right.sum + v.x}{}$$

$$\text{Svar } v.sumofsquares = \frac{v.left.sumofsquares + v.right.sumofsquares + (v.x)^2}{}$$

Opgave 19 (4%)

Angiv worst-case tiden for at finde det $\lceil \sqrt{n} \rceil$ mindste element blandt n elementer, når man *i*) konstruerer en binær min-heap med BUILD-MIN-HEAP og kalder HEAP-EXTRACT-MIN $\lceil \sqrt{n} \rceil$ gang, og *ii*) kalder RANDOMIZED-SELECT.

$$\text{Svar min-heap: } \frac{O(n)}{}$$

$$\text{Svar RANDOMIZED-SELECT: } \frac{O(n^2)}{}$$

Transitionssystem Nedtælling

Konfigurationer: $\{[i, j, s] \mid \text{ikke negative heltal } i, j, s\}$

$$[i, j, s] \triangleright [i, j - 1, s + 1] \quad \text{if } j > 0$$

$$[i, j, s] \triangleright [i - 1, i - 1, s + 1] \quad \text{if } j = 0 \wedge i > 0$$

Opgave 20 (4%)

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

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

Opgave 21 (4%)

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

	Ja	Nej
$\mu(i, k) = i$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(i, k) = s$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(i, k) = i^2 + i + 2j + 2s$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(i, k) = i + j$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(i, k) = i^2 + i + 2j$	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Algoritme ZeroCount(n)

Inputbetingelse : Array $A[1..n]$, hvor $A[k] \in \{0, 1\}$ for $1 \leq k \leq n$

Outputkrav : $i = |\{k \mid 1 \leq k \leq n \wedge A[k] = 0\}|$

Metode : $i \leftarrow 0$

$j \leftarrow n$

$\{I\}$ **while** $i < j$ **do**

if $A[j] = 1$ **then**

$j \leftarrow j - 1$

else

$i \leftarrow i + 1$

$A[j] \leftarrow A[i]$

$A[i] \leftarrow 0$

Opgave 22 (4%)

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

	Ja	Nej
$0 \leq i < j \leq n$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$i = \{k \mid 1 \leq k \leq j \wedge A[k] = 0\} $	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$A[k] = 0$ for $1 \leq k \leq i$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
$A[k] = 1$ for $j \leq k \leq n$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$i + j = n$	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Opgave 23 (4%)

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

	Ja	Nej
$\mu(j, r) = j$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(j, r) = j - i$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
$\mu(j, r) = (n - i)j$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
$\mu(j, r) = n - ij$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$\mu(j, r) = n - j$	<input type="checkbox"/>	<input checked="" type="checkbox"/>

