Opgave 1 (4 %)

	Ja	Nej
$n + n^3$ er $O(4n^2)$?	☐	☒
$n^{2/3}$ er $O(n^{1/3})$?	☐	☒
n^3 er $O(n^2 \cdot \log n)$?	☐	☒
$\sqrt{n}$ er $O((\log n)^2)$?	☐	☒
$7n$ er $\Omega(n^7)$?	☐	☒

Opgave 2 (4 %)

Opskriv følgende funktioner efter stigende orden med hensyn til O-notationen (bemærk at $\log n$ betegner totals logaritmen):

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

Svar: $(\log n)^2$ $\quad \sqrt{2}^{\log n}$ $\quad n/\log n$ $\quad n^2$ $\quad 2^n$

Opgave 3 (4 %)

Antag $f(n), f_1(n), f_2(n), g(n), g_1(n)$ og $g_2(n)$ er heltallige positive ikke-aftagende funktioner. Hvilke af følgende udsagn er sande?

	Ja	Nej
$f(n) = O(g(n)) \implies g(n) = O(f(n))$?	☐	☒
$f(n) = \Omega(g(n)) \implies f(n) = O(g(n))$?	☐	☒
$f(n) = O(g(n)) \implies g(n) = \Omega(f(n))$?	☒	☐
$f_1(n) = O(g_1(n)) \wedge f_2(n) = O(g_2(n)) \implies f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$?	☒	☐
$f_1(n) = O(g_1(n)) \wedge f_2(n) = O(g_2(n)) \implies f_1(n) - f_2(n) = O(g_1(n) - g_2(n))$?	☐	☒

Opgave 4 (4 %)

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

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

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

Algoritme Loop3(n)
  $i \leftarrow 1$
  while $i \leq n$ **do**
    $i \leftarrow 3 * i$

Svar Loop1: $O(n^3)$

Svar Loop2: $O(n^2)$

Svar Loop3: $O(\log n)$

Opgave 5 (4 %)

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

Algoritme Loop1(n)
  if $n \leq 0$ **then**
    return 1
  else
    return Loop1($n-1$) + Loop1($n-1$)

Algoritme Loop2(n)
  $x \leftarrow 0$
  $i \leftarrow 1$
  $s \leftarrow 0$
  while $i \leq n$ **do**
    for $k \leftarrow 1$ **to** s
      for $\ell \leftarrow 1$ **to** k
        $x \leftarrow x + 1$
    $s \leftarrow s + i$
    $i \leftarrow i + 1$

Algoritme Loop3(n)
  $i \leftarrow 1$
  $s \leftarrow 0$
  while $s \leq n$ **do**
    $j \leftarrow 1$
    while $j \leq i$ **do**
      $j \leftarrow 2 * j$
    $s \leftarrow s + i$
    $i \leftarrow i + 1$

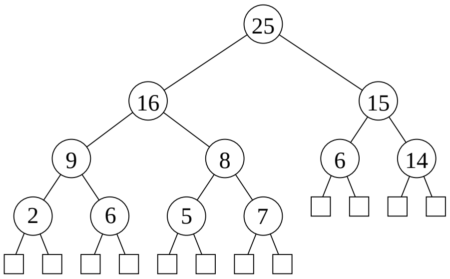
Svar Loop1: $O(2^n)$

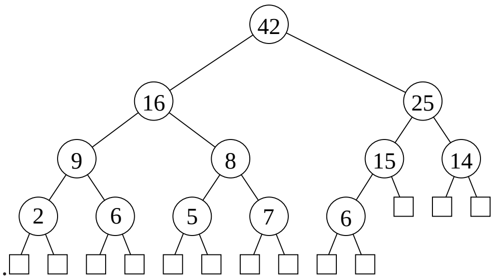
Svar Loop2: $O(n^5)$

Svar Loop3: $O(\sqrt{n} \cdot \log n)$

Opgave 6 (4 %)

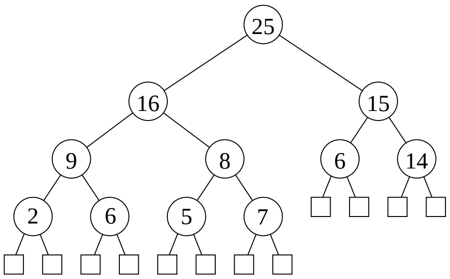
Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 42.

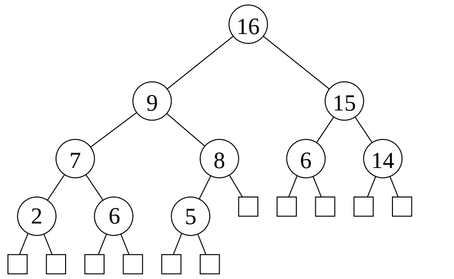




Svar:

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

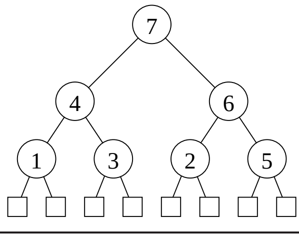




Svar:

Opgave 7 (4 %)

Tegn den binære max-heap efter indsættelse af elementerne 1, 2, 3, 4, 5, 6, 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
2	4	6	8	10	1	3	5	7	9

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

Svar:

Opgave 9 (4 %)

Betragt Counting-Sort anvendt på nedenstående liste af tal fra $\{0, 1, \ldots, 9\}$. Angiv arrayet C efter tallene er blevet sorteret med Counting-Sort.

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

Svar:

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

Opgave 10 (4 %)

Angiv resultatet af at anvende Partition(A,2,11) 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	17	13	4	15	3	5	18	9	11	24	12	14	10	7	22

Svar:

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

Opgave 11 (4 %)

Angiv den asymptotiske tid for quicksort på et input af størrelse n.

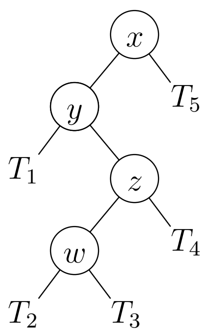
Bedste tid ? Svar: $O(n \log n)$

Værste tid ? Svar: $O(n^2)$

Forventede tid for et tilfældigt input ? Svar: $O(n \log n)$

Opgave 12 (4 %)

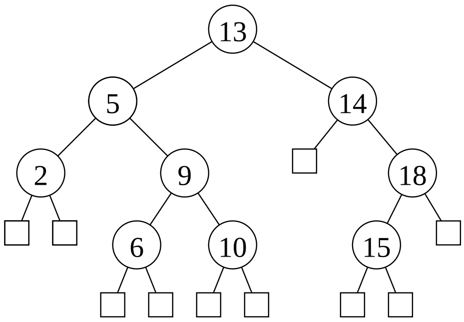
Nedenstående angiver en del af et søgetræ. Angiv x, y, z, og w i sorteret voksende rækkefølge.

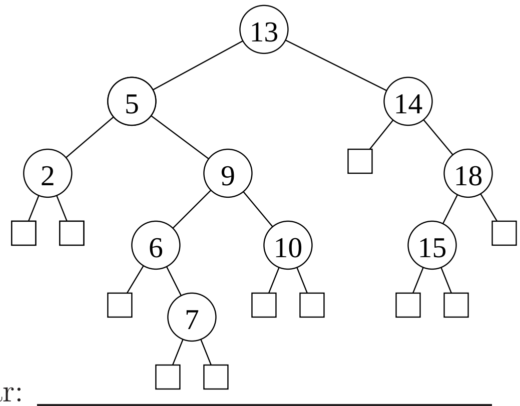


Svar: $y \quad w \quad z \quad x$

Opgave 13 (4 %)

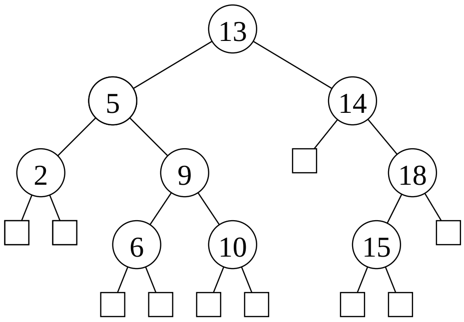
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 7.

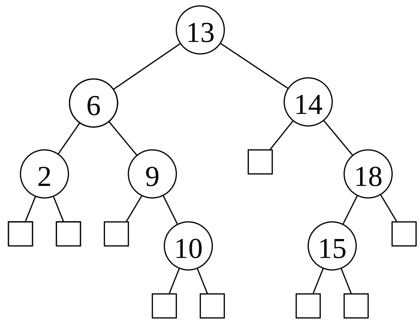




Svar: ____________________

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter slettelse af elementet 5.

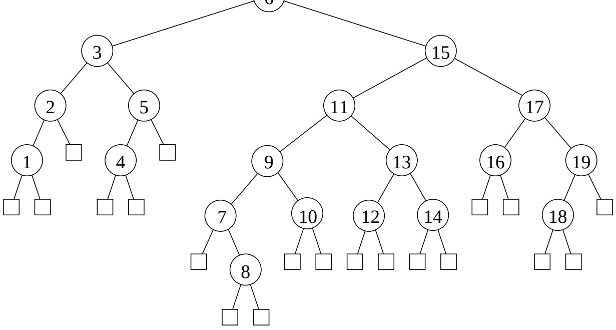


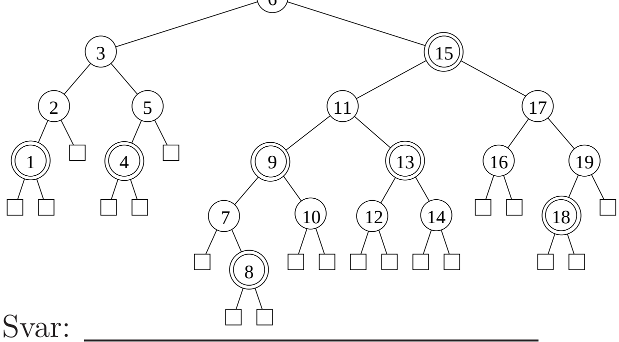


Svar: ____________________

Opgave 14 (4 %)

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

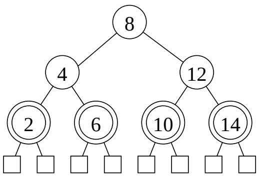


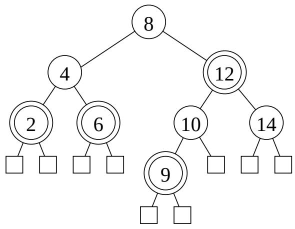


Svar: ____________________

Opgave 15 (4 %)

Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 9.

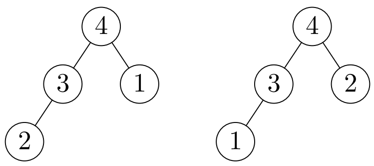




Svar: ____________________

Opgave 16 (4 %)

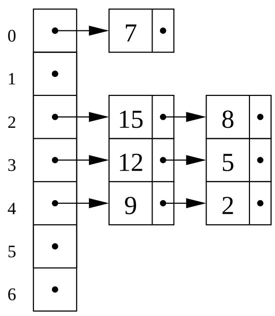
Angiv alle mulige binære max-heaps for mængden $\{1, 2, 3, 4\}$.



Svar: ____________________

Opgave 17 (4 %)

Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hashfunktionen er $h(k) = 2k \bmod 7$ og der indsættes elementerne 5, 2, 7, 9, 8, 15, og 12 i den givne rækkefølge.



Svar: ____________________

Opgave 18 (4 %)

Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 8, 2, 5, 3, 12, 1, og 13 indsættes i den givne rækkefølge, når hashfunktionen er $h(k) = 3k \bmod 10$.

0	1	2	3	4	5	6	7	8	9

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

Svar: ______

Opgave 19 (4 %)

Tegn hvordan en hashtabel der anvender *quadratic probing* ser ud efter at elementerne 9, 3, 14, 0, 1, og 12 indsættes i den givne rækkefølge, når hashfunktionen er $h(k, i) = 2k + 3i + 2i^2 \bmod 11$.

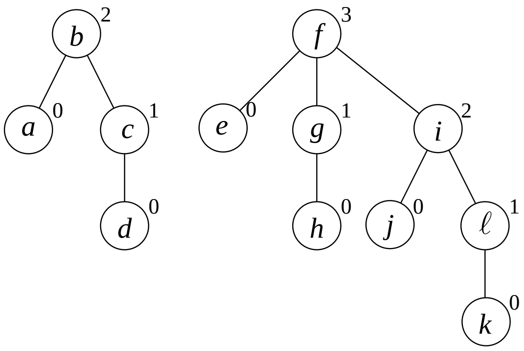
0	1	2	3	4	5	6	7	8	9	10

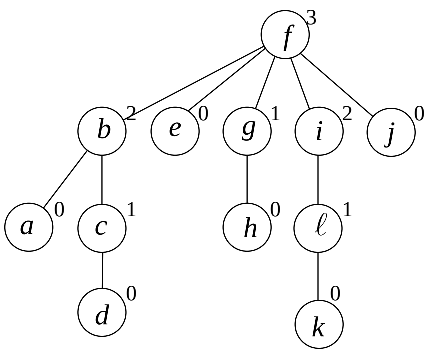
0	1	2	3	4	5	6	7	8	9	10
14	12	1			0	3	9			

Svar: ______

Opgave 20 (4 %)

Angiv den resulterende union-find struktur efter $\textsc{Union}(c, j)$, når der anvendes union-by-rank og stikomprimering (tallene angiver knudernes rang).





Svar: ______

(Opgavesættet fortsætter)

Transitionssystem Count-Down
Konfigurationer: $\{[k, i] \mid \text{heltal } k, i \wedge k \geq 0 \wedge i \geq 0\}$

$[k, i] \triangleright [k - 2^i, i]$ **if** $k \geq 2^i$
$[k, i] \triangleright [k, i - 1]$ **if** $k < 2^i \wedge i > 0$

Opgave 21 (4 %)

For hvert af nedenstående udsagn, angiv om de er en invariant for ovenstående transitionssystem Count-Down. Startkonfigurationen $[k_0, i_0]$ antages at opfylde $2^{i_0+1} > k_0 \geq 2^{i_0}$.

	Ja	Nej
$k \geq 2^i$	☐	☒
$k \geq i$	☐	☒
$k + 2^i = k_0 + 2^{i_0}$	☐	☒
$2^{i+1} > k \geq 2^i$	☐	☒
$k^2 \geq i$	☐	☒

Opgave 22 (4 %)

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

	Ja	Nej
$\mu(k, i) = k + i$	☒	☐
$\mu(k, i) = k + 2^i$	☒	☐
$\mu(k, i) = 2^k + 2^i - i$	☐	☒
$\mu(k, i) = 2^{k+i} - i$	☐	☒
$\mu(k, i) = i \cdot k$	☐	☒

Algoritme Cubed(n)
Inputbetingelse : heltal $n \geq 0$
Outputkrav : $k \leq n^3$
Metode : $i \leftarrow 0$
$j \leftarrow 0$
$k \leftarrow 0$
$\{I\}$ **while** $i < n$ **do**
  $k \leftarrow k + j + j + j + i + i + i + 1$
  $j \leftarrow j + i + i + 1$
  $i \leftarrow i + 1$

Opgave 23 (4 %)

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

	Ja	Nej
$k = n^3$	☐	☒
$j = i^2$	☒	☐
$k = i^3$	☒	☐
$i \leq j \leq k$	☒	☐
$j \leq n$	☐	☒

Opgave 24 (4 %)

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

	Ja	Nej
$\mu(n, i, j, k) = n - i$	☒	☐
$\mu(n, i, j, k) = n^3 - k$	☒	☐
$\mu(n, i, j, k) = n^2 - j$	☒	☐
$\mu(n, i, j, k) = i$	☐	☒
$\mu(n, i, j, k) = (n - i)^2$	☒	☐

Opgave 25 (4 %)

Givet et heltal n, beregner nedenstående algoritme den binære heltals logaritme af n, dvs.

$$\text{intlog}(n) = \max\{i \mid 2^i \leq n \ \wedge \ i \text{ er et heltal}\}$$

For at vise gyldigheden af algoritmen skal I_i og I_r være invarianter omkring variablerne i og r. Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke). Det antages at n ikke kan ændres af algoritmen.

Algoritme intlog(n)
Inputbetingelse : heltal $n \geq 1$
Outputkrav : $i = \text{intlog}(n)$
Metode : $r \leftarrow 1$
$i \leftarrow 0$
$\{I_r \ \wedge \ I_i\}$ **while** $r \leq n$ **do**
  $r \leftarrow r + r$
  $i \leftarrow i + 1$
$i \leftarrow i - 1$

Svar I_i: $r = 2^i$

Svar I_r: $1 \leq r \leq 2n$

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 μ: $2n - r$