Opgave 1 (4 %)

	Ja	Nej
n er $O(n^2)$?	☒	☐
$n \cdot \log n$ er $O(n^{3/2})$?	☒	☐
n^2 er $O((\log n)^3)$?	☐	☒
$(2/3)^n$ er $O(\log n)$?	☒	☐
$23n^2 + 7n$ er $O(30n)$?	☐	☒

Opgave 2 (4 %)

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

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

Svar: $(\log n)^7 \quad n^{1/7} \quad 7n \quad n^7 \quad 7^n$

Opgave 3 (4 %)

Betragt varianten af den binære max-heap, hvor hver knude har d børn istedet for to, og hvor $d \geq 2$ er en parameter. En max-heap har nu højde $O(\log_d n)$.

Hvad er tiden for Insert og Extract-Max udtrykt som funktion af n og d.

Insert ? Svar: $O(\log_d n)$

Extract-Max ? Svar: $O(d \cdot \log_d n)$

Hvis vi sætter $d = \Theta(\sqrt{n})$, hvad bliver tiden for Insert og Extract-Max udtrykt som funktion af n ?

Insert ? Svar: $O(1)$

Extract-Max ? Svar: $O(\sqrt{n})$

Opgave 4 (4 %)

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

Algoritme Loop1(n)
  for $i = 1$ **to** n
    for $j = i$ **to** n
      for $k = i$ **to** j
        $s = s + 1$

Algoritme Loop2(n)
  $s = 0$
  while $n > 1$
    for $i = 1$ **to** n
      $s = s + 1$
    $n = \lfloor n/2 \rfloor$

Algoritme Loop3(n)
  for $i = 1$ **to** n
    $s = 1$
    while $s \leq i$
      $s = 2 * s$

Svar Loop1: $O(n^3)$

Svar Loop2: $O(n)$

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

Opgave 5 (4 %)

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

Algoritme Loop1(n)
  $i = 1$
  while $i < n$
    $i = i * n$

Algoritme Loop2(n)
  $i = 1$
  while $i < n$
    $i = i * 2$

Algoritme Loop3(n)
  for $i = 1$ **to** n
    $s = 0$
    while $s < n$
      $s = s + i$

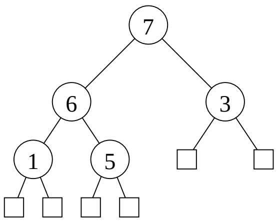
Svar Loop1: $O(1)$

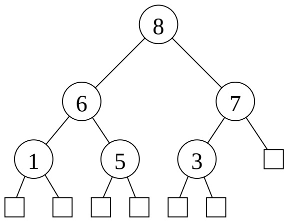
Svar Loop2: $O(\log n)$

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

Opgave 6 (4 %)

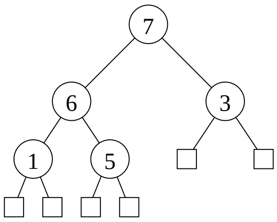
Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 8.

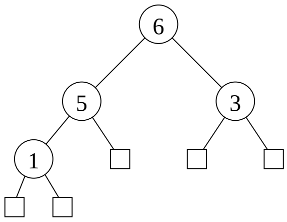




Svar: ______________________

Tegn hvordan nedenstående binære max-heap ser ud efter en Heap-Extract-Max operation.

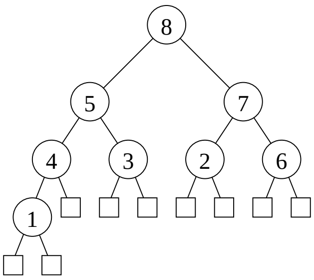




Svar: ______________________

Opgave 7 (4 %)

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

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

Svar: ______________________

Opgave 9 (4 %)

Betragt Radix-Sort anvendt på nedenstående liste af tal ($d = 4$, $k = 6$). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter de *tre* mindst betydende cifre.

1453 4213 5123 4311 3213 3311

Svar: 5123 4213 3213 4311 3311 1453

Opgave 10 (4 %)

Angiv resultatet af at anvende Partition(A,3,11) på nedenstående array.

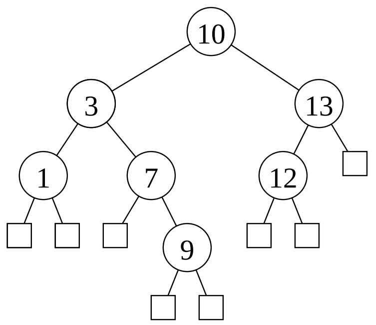
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	8	16	1	6	2	4	13	17	15	3	5	18	9	11	24	12

Svar:

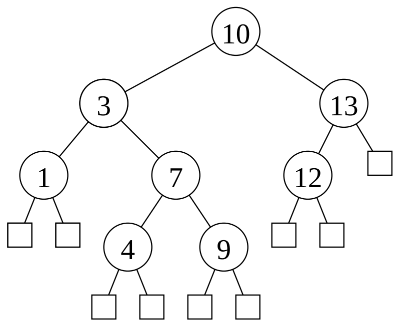
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
8	16	**1**	**2**	**4**	**3**	**5**	**17**	**15**	**6**	**13**	18	9	11	24	12

Opgave 11 (4 %)

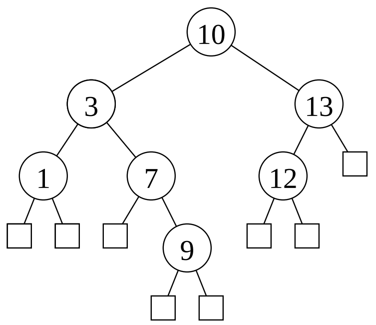
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 4.



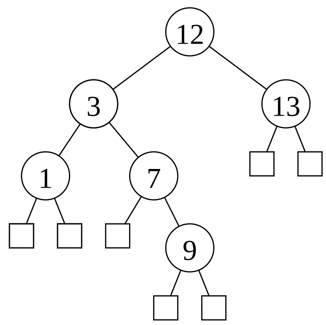
Svar:



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

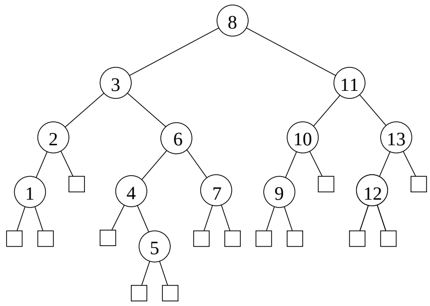


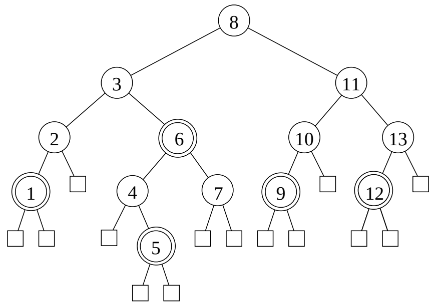
Svar:



Opgave 12 (4 %)

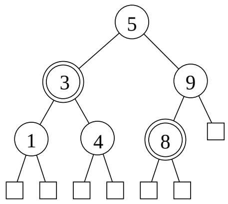
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.

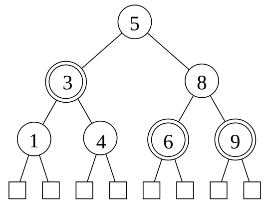


Svar: 

Opgave 13 (4 %)

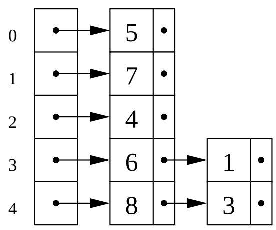
Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 6.



Svar: 

Opgave 14 (4 %)

Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hashfunktionen er $h(k) = 3k \bmod 5$ og der indsættes elementerne 7, 5, 1, 3, 4, 6 og 8 i den givne rækkefølge.

Svar: 

Opgave 15 (4 %)

Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 1, 2, 3, 8, 10 og 9 indsættes i den givne rækkefølge, når hashfunktionen er $h(k) = 4k \bmod 7$.

0	1	2	3	4	5	6

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

Svar: ____________________

Opgave 16 (4 %)

Tegn hvordan en hashtabel der anvender *double hashing* ser ud efter at elementerne 0, 4, 2 og 8 indsættes i den givne rækkefølge, når hashfunktionen er

$$h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod 11$$

$$h_1(k) = 3k \bmod 13 \qquad h_2(k) = 1 + (4k \bmod 9)$$

0	1	2	3	4	5	6	7	8	9	10

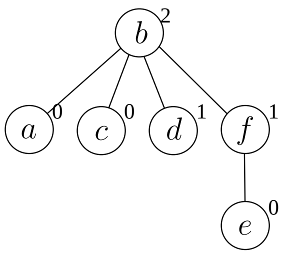
0	1	2	3	4	5	6	7	8	9	10
0	4					2	8			

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(d,a)
union(e,f)
union(c,e)



Svar: ____________________

Opgave 18 (4 %)

Betragt union-find strukturer hvor der anvendes linking-by-rank. Hvad er tiden for FIND på en mængde med n elementer – med og uden stikomprimering ?

Uden stikomprimering ? Svar = $O(\log n)$

Med stikomprimering ? Svar = $O(\log n)$

Opgave 19 (4 %)

For en liste af punkter $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, hvor $x_1 < x_2 < \cdots < x_n$, definerer vi punkterne til at være *monotont voksende* hvis der gælder

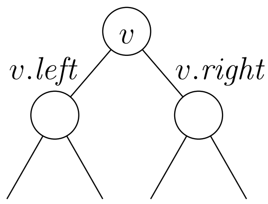
$$y_1 < y_2 < \cdots < y_n$$

F.eks. er punkterne

$$(1, 3), (4, \underline{7}), (6, \underline{5}), (8, 9)$$

ikke monotont voksende da $7 > 5$.

Betragt et søgetræ hvor hver knude v gemmer et punkt, og hvor punkterne er sorteret fra venstre-mod-højre efter deres x-værdi. I hver knude gemmes ud over $v.x$ og $v.y$ også $v.ymin$ og $v.ymax$, som er hhv. den mindste og største y-værdi i v's undertræ, og en boolsk variabel $v.inc$ der angiver om listen af punkter i v's undertræ er monotont voksende. 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).



Svar $v.ymin$ = $\min\{v.left.ymin, v.y, v.right.ymin\}$

Svar $v.ymax$ = $\max\{v.left.ymax, v.y, v.right.ymax\}$

Svar $v.inc$ = $v.left.inc \wedge v.right.inc \wedge v.left.ymax < v.y \wedge v.y < v.right.ymin$

Transitionssystem Frem-og-Tilbage
Konfigurationer: $\{[i,k] \mid \text{heltal } i,k \wedge i \geq 0 \wedge k \geq 0\}$

$[i,k] \triangleright [i-3, k+2]$ **if** $i \geq 3$
$[i,k] \triangleright [i+1, k-1]$ **if** $k \geq 1$

Opgave 20 (4 %)

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

	Ja	Nej
$i + k = 2n$	☐	☒
$i \leq 2n$	☒	☐
$i + k \leq 2n$	☒	☐
$2i + 3k \leq 5n$	☒	☐
$i \geq k$	☐	☒

Opgave 21 (4 %)

For hver af nedenstående funktioner, angiv om de er en termineringsfunktion for ovenstående transitionssystem Frem-og-Tilbage.

	Ja	Nej
$\mu(i,k) = 2i + 3k$	☐	☒
$\mu(i,k) = i + k$	☐	☒
$\mu(i,k) = 3i + 4k$	☒	☐
$\mu(i,k) = 3i + 2k$	☐	☒
$\mu(i,k) = \min\{i,k\}$	☐	☒

Algoritme Power(n, x)
Inputbetingelse : heltal n og x, hvor $x \geq 2$ og $n \geq 1$
Outputkrav : $r = x^n$
Metode : $r \leftarrow 1$;
$p \leftarrow n$;
$y \leftarrow x$;
$\{I\}$ **while** $p > 0$ **do**
  if p ulige **then**
    $r \leftarrow r * y$;
    $p \leftarrow p - 1$
  else
    $y \leftarrow y * y$;
    $p \leftarrow p/2$

Opgave 22 (4 %)

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

	Ja	Nej
$x^n = y^p$	☐	☒
$x^n = r \cdot y^p$	☒	☐
$r = x^n$	☐	☒
$r \cdot x^n = y^p$	☐	☒
$p > 0$	☐	☒

Opgave 23 (4 %)

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

	Ja	Nej
$\mu(n, x, r, p, y) = x^n - r$	☐	☒
$\mu(n, x, r, p, y) = y^p - r$	☐	☒
$\mu(n, x, r, p, y) = p$	☒	☐
$\mu(n, x, r, p, y) = x^n - y^p$	☐	☒
$\mu(n, x, r, p, y) = 2x^n - r - y$	☒	☐

(Opgavesættet fortsætter)

Opgave 24 (4 %)

Givet et positivt heltal n, så beregner nedenstående algoritme summen

$$\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}$$

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

Algoritme Sum(n)
Inputbetingelse : positivt heltal $n \geq 1$
Outputkrav : $r = \sum_{j=1}^{n} j^2$
Metode : $i \leftarrow 1$;
$p \leftarrow 1$;
$r \leftarrow 1$;
$\{I_i \ \wedge \ I_p \ \wedge \ I_r\}$ **while** $i < n$ **do**
$p \leftarrow p + i + i + 1$;
$r \leftarrow r + p$;
$i \leftarrow i + 1$

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

Svar I_p: $p = i^2$

Svar I_r: $r = \sum_{j=1}^{i} j^2$

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 μ: $n - i$

Opgave 25 (4 %)

For at argumentere for at tiden for at rebalancere et rød-sort træ efter indsættelsen af en ny knude er amortiseret $O(1)$ skal man anvende en potentiale funktion. Angiv en sådan potentiale funktion.

Svar Φ : # røde knuder