Opgave 1 (4 %)

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

Opgave 2 (4 %)

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

$$n^7$$
$$n^{\frac{1}{7}}$$
$$7/n$$
$$\tfrac{1}{7}n$$
$$7n \log n$$

Svar: $7/n$ $n^{\frac{1}{7}}$ $\frac{1}{7}n$ $7n \log n$ n^7

Opgave 3 (4 %)

I denne opgave ønsker vi at sortere n elementer, hvor hvert element enten er 0 eller 1. Angiv "worst case" og "best case" tiderne for hver af nedenstående algoritmer.

	Worst case	Best case
CountingSort	$O(n)$	$O(n)$
InsertionSort	$O(n^2)$	$O(n)$
MergeSort	$O(n \log n)$	$O(n \log n)$
QuickSort	$O(n^2)$	$O(n^2)$

(Opgavesættet fortsætter)

Opgave 4 (4 %)

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

```
Algoritme Loop1(n)
  s = 0
  for i = 1 to n
    for j = 1 to i * i
      s = s + 1
```

```
Algoritme Loop2(n)
  s = 0
  for i = 0 to n
    for j = 0 to n
      s = s + 1
```

```
Algoritme Loop3(n)
  i = 0
  j = n
  while i ≤ j
    i = i + 1
    j = j - 1
```

Svar Loop1: $O(n^3)$

Svar Loop2: $O(n^2)$

Svar Loop3: $O(n)$

Opgave 5 (4 %)

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

```
Algoritme Loop1(n)
  i = n
  while i > 0
    if i ulige
      i = i - 1
    else
      i = i/2
```

```
Algoritme Loop2(n)
  s = 1
  i = 1
  while i ≤ n
    for j = 1 to i
      s = s + 1
    i = i * 2
```

```
Algoritme Loop3(n)
  i = 1
  while i * i ≤ n
    i = i + i
```

Svar Loop1: $O(\log n)$

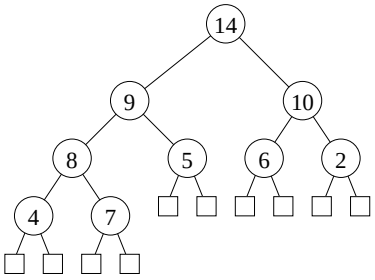
Svar Loop2: $O(n)$

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

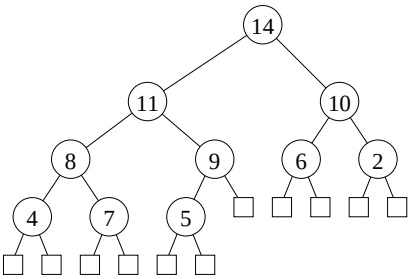
(Opgavesættet fortsætter)

Opgave 6 (4 %)

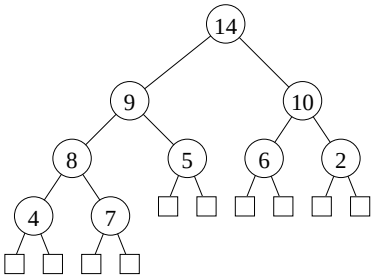
Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 11.



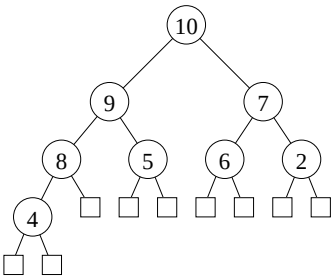
Svar:



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



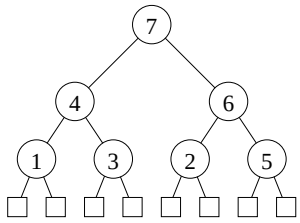
Svar:



Opgave 7 (4 %)

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

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

Svar:

(Opgavesættet fortsætter)

Opgave 9 (4 %)

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

4411 1034 3211 0434 5304 2111

Svar: 5304 4411 3211 2111 1034 0434

Opgave 10 (4 %)

Angiv resultatet af at anvende PARTITION(A,5,12) på nedenstående array.

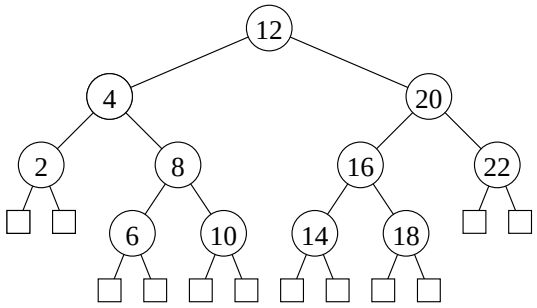
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	10

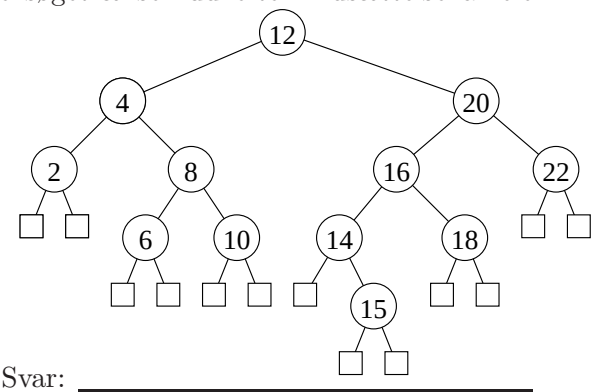
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	15	14	13	12	**4**	**10**	**9**	**8**	**7**	**6**	**5**	**11**	3	2	10

Svar:

Opgave 11 (4 %)

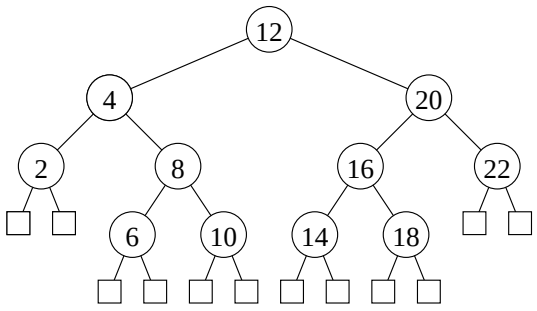
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 15.

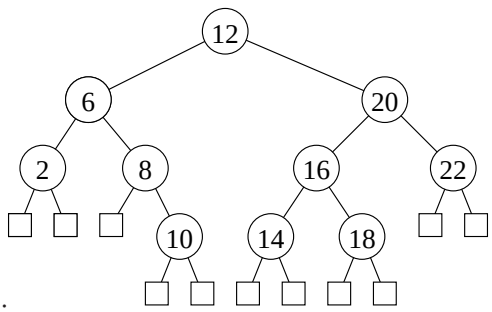




Svar:

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



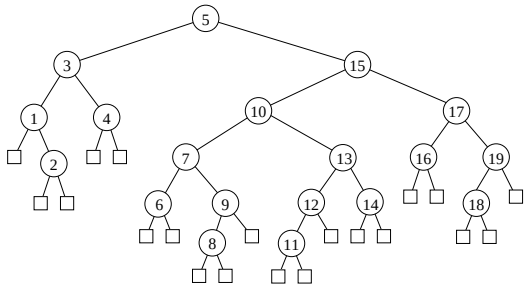


Svar:

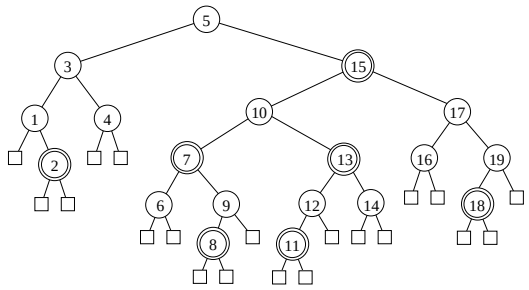
(Opgavesættet fortsætter)

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.

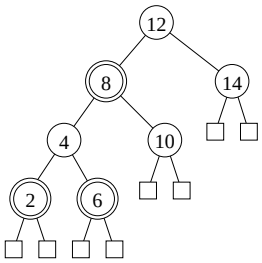


Svar:

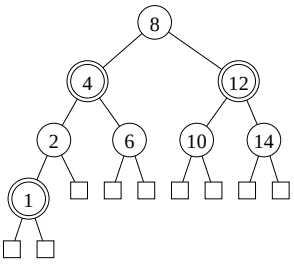


Opgave 13 (4 %)

Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 1.



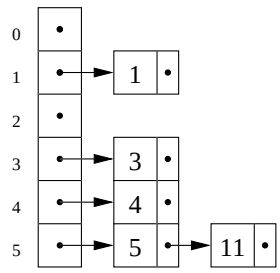
Svar:



Opgave 14 (4 %)

Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hash-funktionen er $h(k) = k \bmod 6$ og der indsættes elementerne 3, 1, 4, 11, og 5 i den givne rækkefølge.

Svar:

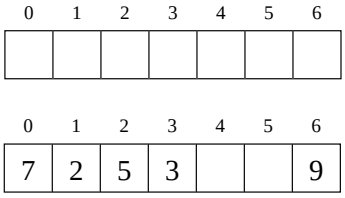


(Opgavesættet fortsætter)

Opgave 15 (4 %)

Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 9, 7, 2, 5, og 3 indsættes i den givne rækkefølge, når hashfunktionen er

$$h(k) = 3k \bmod 7 \, .$$

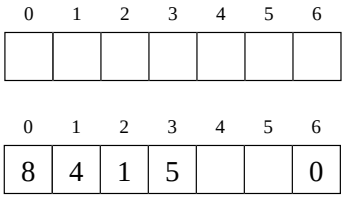


Svar: _____

Opgave 16 (4 %)

Tegn hvordan en hashtabel der anvender *kvadratisk probing* ser ud efter at elementerne 1, 4, 5, 8, og 0 indsættes i den givne rækkefølge, når hashfunktionen er

$$h(k, i) = ((2k \bmod 9) + i + i^2) \bmod 7 \, .$$



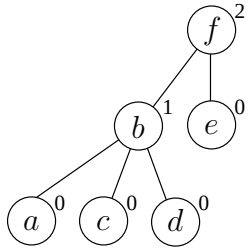
Svar: _____

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

Svar: _____



(Opgavesættet fortsætter)

Opgave 18 (4 %)

Hvad er det minimale antal elementer der kan være i en union-find struktur, når der anvendes union-by-rank og stikomprimering, og når roden har rang r? Angiv resultatet som funktion af r.

Svar: 2^r

Hvad er den maksimale dybde en union-find struktur med n elementer kan have, når der anvendes union-by-rank og stikomprimering? Angiv resultatet som funktion af n.

Svar: $\lfloor \log n \rfloor$

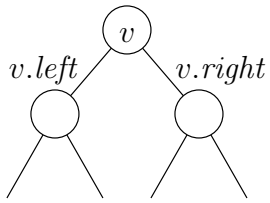
Hvad er det maksimale antal børn en rod med rang r kan have i en union-find struktur med n elementer, når der anvendes union-by-rank og stikomprimering? Angiv resultatet som funktion af n og r.

Svar: $n - 1$

Opgave 19 (4 %)

Betragt et søgetræ hvor hver knude v gemmer et element $v.x$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i v også $v.size$ og $v.avg$, som er hhv. antal knuder i v's undertræ, og gennemsnittet af alle elementerne i v's undertræ (dvs. summen af alle elementer i v's undertræ divideret med antal elementer i undertræet).

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 $v.size$ = $v.left.size + 1 + v.right.size$

Svar $v.avg$ = $(v.left.avg \cdot v.left.size + v.x + v.right.avg \cdot v.right.size)/v.size$

(Opgavesættet fortsætter)

Transitionssystem Multiply
Konfigurationer: $\{[m,n,p] \mid \text{heltal } m,n,p \ \wedge \ m \geq 0 \ \wedge \ n \geq 0 \ \wedge \ p \geq 0\}$

$[m,n,p] \rhd [m-1,n,p+n]$ **if** $m > 0 \ \wedge \ m$ ulige
$[m,n,p] \rhd [m/2,n \cdot 2,p]$ **if** $m > 0 \ \wedge \ m$ lige

Opgave 20 (4 %)

For hvert af nedenstående udsagn, angiv om det er en invariant for ovenstående transitionssystem Multiply. Startkonfigurationen antages at være $[m_0,n_0,0]$, hvor $m_0 \geq 0$ og $n_0 \geq 0$.

	Ja	Nej
$p = 0$	☐	☒
$m \cdot n = p$	☐	☒
$m \cdot n = p + m_0 \cdot n_0$	☐	☒
$m \cdot n + p = m_0 \cdot n_0$	☒	☐
$m_0 \cdot n_0 - m \cdot n \geq 0$	☒	☐

Opgave 21 (4 %)

For hver af nedenstående funktioner, angiv om den er en termineringsfunktion for ovenstående transitionssystem Multiply.

	Ja	Nej
$\mu(m,n,p) = mn - p$	☐	☒
$\mu(m,n,p) = mn$	☐	☒
$\mu(m,n,p) = m$	☒	☐
$\mu(m,n,p) = 2m_0n_0 - p - n$	☐	☒
$\mu(m,n,p) = m - n$	☐	☒

(Opgavesættet fortsætter)

Algoritme Loglog(n)
Inputbetingelse : Heltal $n \geq 2$
Outputkrav : $r = \max\{i \mid 2^{2^i} \leq n\} = \lfloor \log \log n \rfloor$
Metode : $r \leftarrow 0$;
$p \leftarrow 2$;
$\{I\}$ **while** $p \cdot p \leq n$ **do**
$r \leftarrow r + 1$;
$p \leftarrow p \cdot p$

Opgave 22 (4 %)

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

	Ja	Nej
$r < p$	☒	☐
$p \cdot p \leq n$	☐	☒
$p = 2^{2^r}$	☒	☐
$r = \lfloor \log \log n \rfloor$	☐	☒
$n - p \geq 0$	☒	☐

Opgave 23 (4 %)

For hver af nedenstående funktioner, angiv om den er en termineringsfunktion for ovenstående algoritme Loglog.

	Ja	Nej
$\mu(n, r, p) = p$	☐	☒
$\mu(n, r, p) = n - p \cdot p$	☐	☒
$\mu(n, r, p) = n - p$	☒	☐
$\mu(n, r, p) = n - r$	☒	☐
$\mu(n, r, p) = r - \log \log p$	☐	☒

(Opgavesættet fortsætter)

Opgave 24 (4 %)

Lad $C[1..n]$ være et array, hvor $C[i]$ angiver en farve (repræsenteret ved et heltal), og lad max-monochrome(C) betegne længden af det længste monochrome delarray, dvs. delarray hvor alle indgange har samme farve. Nedenstående algoritme beregner max-monochrome(C). For at vise gyldigheden af algoritmen skal $I_{\ell,r}$ og I_m være invarianter omkring variablerne ℓ, r, og m. Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke).

Algoritme MaxMonochrome($C[1..n]$)
Inputbetingelse : Array $C[1..n]$ med n heltal, hvor $n \geq 1$
Outputkrav : $m =$ max-monochrome(C)
Metode :
```
m ← 1;
ℓ ← 1;
r ← 1;
{I} while r < n do
        if C[r+1] = C[r] then
            r ← r + 1;
            m ← max{m, r − ℓ + 1}
        else
            r ← r + 1;
            ℓ ← r
```

Svar $I_{\ell,r}$: $1 \leq \ell \leq r \leq n \wedge C[\ell] = C[\ell+1] = \cdots = C[r] \wedge (\ell = 1 \vee C[\ell-1] \neq C[\ell])$

Svar I_m: $m =$ max-monochrome($C[1..r]$)

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: $n - r$

Opgave 25 (4 %)

Betragt et array $A[0..N-1]$ af længde N, hvor hver indgang er enten 0 eller 1. A opfattes som et binært tal med værdien n, hvor $n = \sum_{i=0}^{N-1} 2^i \cdot A[i]$. Operationen Inc($A$) øger værdien af n med 1 ved at flippe $A[0], A[1], ...$ indtil det første $A[i]$ flippes fra 0 til 1. Hvis alle indgange i A indeholdt 1 (dvs. $n = 2^N - 1$ før Inc(A) udføres), så kopieres A over i nyt array af dobbelt størrelse $2N$, hvor alle $A[i] = 0$ bort set fra $A[N] = 1$. Med en passende potentialefunktion kan man argumentere for at Inc tager amortiseret $O(1)$ tid. Angiv en sådan potentialefunktion.

Svar $\Phi =$ $N - \lfloor \log n \rfloor + |\{i \mid A[i] = 1\}|$ eller blot $|\{i \mid A[i] = 1\}|$