DATALOGISK INSTITUT, AARHUS UNIVERSITET

Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 1 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 13 (tretten)
Eksamensdag: Torsdag den 23. marts 2006, kl. 9.00-11.00
Eksamenslokale: Trøjborg, Willemoesgade 15, Århus N, 8200 Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger og notater)
Materiale der udleveres til eksaminanden:

Årskort ____________________

Navn ____

Skriftlig Eksamen
Algoritmer og Datastrukturer 1 (2003-ordning)

Datalogisk Institut
Aarhus Universitet

Torsdag den 23. marts 2006, kl. 9.00-11.00

Dette eksamenssæt består af en kombination af små skriftlige opgaver og multiple-choice-opgaver. Opgaverne besvares på opgaveformuleringen **som afleveres**.

For hver opgave er angivet opgavens andel af det samlede eksamenssæt.

For multiple-choice-opgaver gælder følgende. Hvert delspørgsmål har præcist et svar. For hvert delspørgsmål, kan du vælge ét svar ved at afkrydse den tilsvarende rubrik. Et multiple-choice-delspørgsmål bedømmes som følgende:

- Hvis du sætter kryds ved det rigtige svar, får du 1 point.
- Hvis du ikke sætter nogen krydser, får du 0 point.
- Hvis du sætter kryds ved et forkert svar, får du $-\frac{1}{k-1}$ point, hvor k er antal svarmuligheder.

For en multiple-choice-opgave med vægt $v\,\%$ og med n delspørgsmål, hvor du opnår samlet s point, beregnes din besvarelse af multiple-choice-opgaven som:

$$\max\left\{0\,,\,\frac{s}{n}\right\}\,\cdot\,v\,\%$$

Opgave 1 (4 %)

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

Opgave 2 (4 %)

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

$$n^4$$
$$n^3 \cdot (\log n)^3$$
$$4^n / \log n$$
$$1/\sqrt{n}$$
$$\sqrt{n} \cdot \log n$$

Svar: ____

Opgave 3 (4 %)

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

```
Algoritme Loop1(n)
  x ← 0
  for i ← 1 to n do
    for j ← i to n do
      for k ← i to j do
        x ← x + 1
```

```
Algoritme Loop2(n)
  for i ← 1 to n do
    j ← 1
    while j ≤ n do
      j ← j * 2
```

```
Algoritme Loop3(n)
  i ← 1
  j ← n
  while j ≥ 0 do
    j ← j − i
    i ← i + 1
```

Svar Loop1: ____________________

Svar Loop2: ____________________

Svar Loop3: ____________________

Opgave 4 (4 %)

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

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

Algoritme Loop2(n)
  $i \leftarrow n$
  $x \leftarrow 0$
  while $i \geq 1$ **do**
    for $j \leftarrow 1$ **to** i **do**
      $x \leftarrow x + 1$
    $i \leftarrow \lfloor i/2 \rfloor$

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

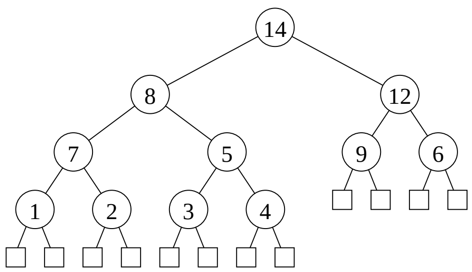
Svar Loop1: ______________________

Svar Loop2: ______________________

Svar Loop3: ______________________

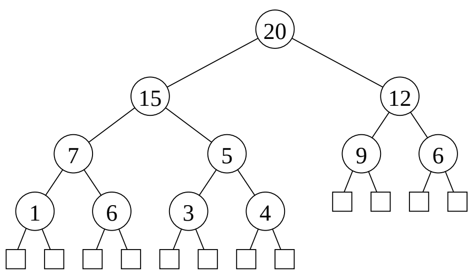
Opgave 5 (4 %)

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 8, 5, 2, 6, 4, 10, 9, 7 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	14	15	16	17
5	2	16	13	8	6	19	15	3	7	9	20	4	12	11	1	10

Svar: ____

Opgave 8 (4 %)

Angiv for hver af nedenstående sorterings-algoritmer deres best-case og worst-case udførselstid for at sortere n elementer.

Algoritme	Worst-case	Best-case
InsertionSort	______________	______________
MergeSort	______________	______________
HeapSort	______________	______________
QuickSort	______________	______________

Opgave 9 (4 %)

Hvad er udførselstiden for insertion-sort på følgende sekvenser, hvor det antages at n er lige.

a) $1, 2, 3, 4, \ldots, n-1, n$
b) $n, n-1, \ldots, 4, 3, 2, 1$
c) $2, 1, 4, 3, 6, 5, \ldots, n, n-1$

Svar a): ____________________

Svar b): ____________________

Svar c): ____________________

Opgave 10 (4 %)

Betragt radix-sort anvendt på nedenstående liste af tal ($d = 5$, $k = 10$). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter de tre mindst betydende cifre.

43118 73144 89123 22118 38144 56743

Svar: ____

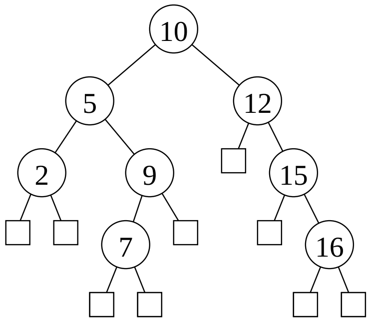
Opgave 11 (4 %)

Angiv en sekvens af n insert operationer i et ubalanceret søgetræ som tager tid $\Theta(n^2)$, når man starter med et tomt søgetræ. Angiv både sekvensen og det resulterende søgetræ.

Svar sekvens: ____________________ Svar træ: ____________________

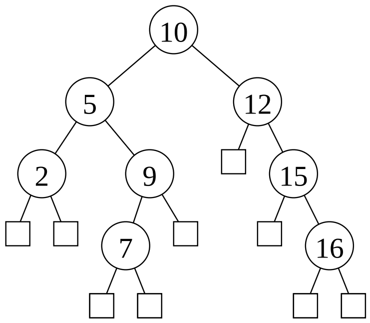
Opgave 12 (4 %)

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 8.



Svar: ______________________________

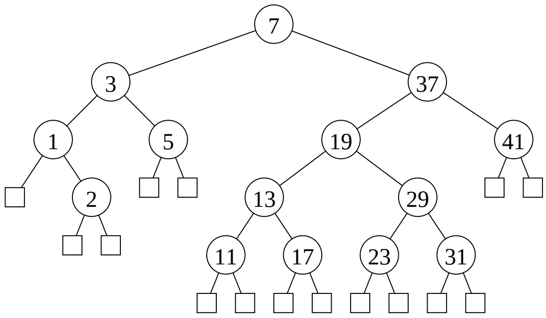
Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter slettelse af elementet 12.



Svar: ______________________________

Opgave 13 (4 %)

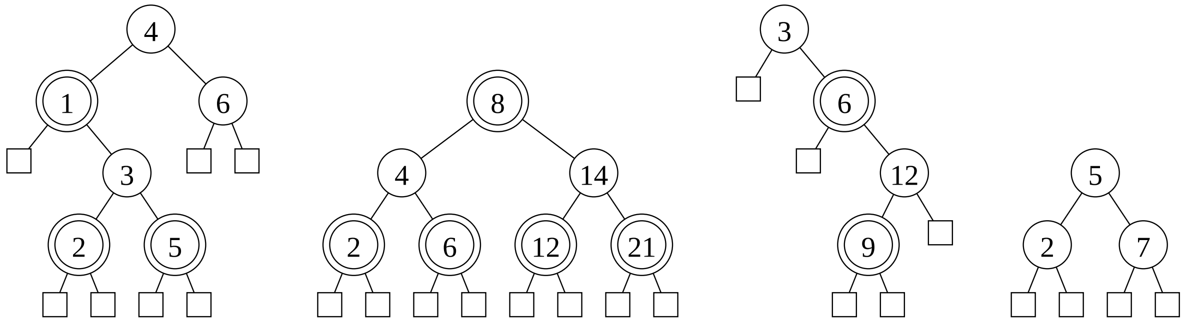
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.



Svar: ______________________________

Opgave 14 (4 %)

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

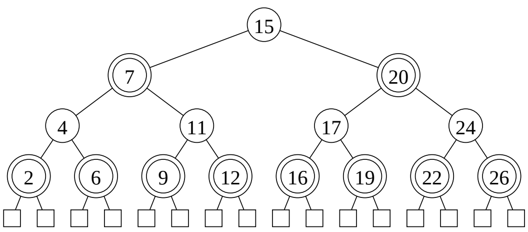


a) b) c) d)

	Rød-sort søgetræ	Søgetræ, men ikke rød-sort	Ikke et søgetræ
a)	☐	☐	☐
b)	☐	☐	☐
c)	☐	☐	☐
d)	☐	☐	☐

Opgave 15 (4 %)

Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 10.



Svar: ______________________

(Opgavesættet fortsætter)

Opgave 16 (4 %)

Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hashfunktionen er $h(k) = k * 3 \bmod 7$ og der indsættes elementerne 3, 8, 2, 9, 1, 10, og 15 i den givne rækkefølge.

Svar: ______________________________

Opgave 17 (4 %)

Nedenstående er en hashtabel hvor der er anvendt *linear probing*. Den anvendte hashfunktion er $h(k) = 2 \cdot k \bmod 17$. Tegn hvordan hashtabellen ser ud efter at elementer 12, 2, 19, 11, 5 er indsat i den givne rækkefølge.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	9					3	20					6				

Svar: ______________________________

Opgave 18 (4 %)

Nedenstående er en hashtabel hvor der er anvendt *quadratic probing*. Den anvendte hashfunktion er

$$h(k,i) = h'(k) + i + 3 \cdot i^2 \mod 11$$

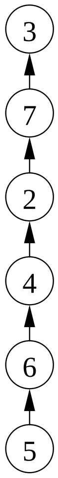
hvor $h'(k) = k \bmod 11$.
Tegn hvordan hashtabellen ser ud efter at $k = 4$ og $k = 14$ indsættes i den givne rækkefølge.

0	1	2	3	4	5	6	7	8	9	10
			3			25	36			

Svar: ______________________________

Opgave 19 (4 %)

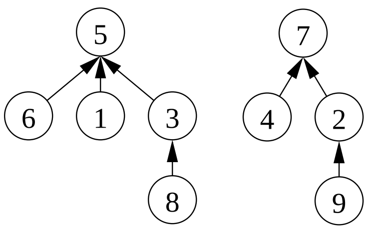
Tegn hvordan nedenstående union-find datastruktur ser ud efter FIND(4), når der anvendes stikomprimering.



Svar: ______________________________

Opgave 20 (4 %)

Tegn hvordan nedenstående union-find datastruktur ser ud efter UNION(8,4) efterfulgt af FIND(9), når der anvendes både union-by-size og stikomprimering.



Svar: ______________________________

Transitionssystem Up-and-Down
Configurations: $\{[i,j] \mid i,j \geq 0\}$

$[i,j] \triangleright [i,j+i]$ **if** $j+i \leq 100$
$[i,j] \triangleright [i+1,j-1]$ **if** $j > 0$

Opgave 21 (4 %)

For hver af nedenstående udsagn, angiv om de er en invariant for ovenstående transitionssystem Up-and-Down. Startkonfigurationen antages at være $[1,0]$.

	Ja	Nej
$i \geq 1$	☐	☐
$j \leq i$	☐	☐
$0 \leq j \leq 100$	☐	☐
$j+i \leq 200$	☐	☐
$i \leq 200$	☐	☐

Opgave 22 (4 %)

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

	Ja	Nej
$\mu(i,j) = i - j$	☐	☐
$\mu(i,j) = j$	☐	☐
$\mu(i,j) = i + j$	☐	☐
$\mu(i,j) = -2i - j$	☐	☐
$\mu(i,j) = 500 - 2i - j$	☐	☐

Algoritme Loop(n)
Inputbetingelse : heltal $n \geq 4$
Outputkrav : –
Metode : $i \leftarrow 4$;
$j \leftarrow 1$;
$\{I\}$ **while** $i < n$ **do**
    $i \leftarrow i * i$;
    $j \leftarrow j + 1$

Opgave 23 (4 %)

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

	Ja	Nej
$4 \leq i \leq n$	☐	☐
$j \leq n$	☐	☐
$0 \leq j \leq i$	☐	☐
$i = 2^j$	☐	☐
$i = 2^{2^j}$	☐	☐

Opgave 24 (4 %)

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

	Ja	Nej
$\mu(i,j,n) = i + j$	☐	☐
$\mu(i,j,n) = n - i$	☐	☐
$\mu(i,j,n) = n - j$	☐	☐
$\mu(i,j,n) = j - i$	☐	☐
$\mu(i,j,n) = n - \sqrt{i}$	☐	☐

Opgave 25 (4 %)

Nedenstående algoritme beregner $r = n^2$. For at vise gyldigheden af algoritmen skal I_n, I_i og I_s være invarianter omkring variablerne n, i og s. Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke).

Algoritme Kvadrat(n)
Inputbetingelse : heltal $n \geq 1$
Outputkrav : $r = n^2$
Metode : $s = 1$;
$i = 1$;
$\{I_n \wedge I_i \wedge I_s\}$ **while** $i < n$ **do**
$i \leftarrow i + 1$;
$s \leftarrow s + i$;
$r = 2 * s - i$

Svar I_n: ______________________________

Svar I_i: ______________________________

Svar I_s: ______________________________

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: ______________________________