DATALOGISK INSTITUT, AARHUS UNIVERSITET

Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 1 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 12 (tolv)
Eksamensdag: Fredag den 13. august 2010, kl. 9.00-11.00
Eksamenslokale: Åbogade 36, 8200 Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger og notater). Computer må ikke medbringes.
Materiale der udleveres til eksaminanden:

Årskort _____

Navn _____

Skriftlig Eksamen
Algoritmer og Datastrukturer 1 (2003-ordning)

Datalogisk Institut
Aarhus Universitet

Fredag den 13. august 2010, 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:

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

(Opgavesættet fortsætter)

Opgave 1 (4 %)

	Ja	Nej
n^2 er $\Omega(n)$?	☐	☐
n^3 er $O(2^n)$?	☐	☐
$7n$ er $O(8 \log n)$?	☐	☐
$5 + 7$ er $O(n)$?	☐	☐
$\sqrt{n}$ er $O(n^{1/3})$?	☐	☐

Opgave 2 (4 %)

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

$$(\log n)^7$$
$$n^{1/2}$$
$$2^n/n^3$$
$$2 \log n$$
$$n^2/\log n$$

Svar: _____

Opgave 3 (4 %)

I det følgende angiver f og g positive ikke-aftagende funktioner. Hvilke af følgende udsagn medfører at $f(n)$ er $O(g(n))$?

	Ja	Nej
Der findes $n \in \mathbb{N}$ så $f(n) \leq g(n)$	☐	☐
For alle $n \in \mathbb{N}$ er $f(n) \leq g(n)$	☐	☐
For alle $N \in \mathbb{N}$ findes $n \geq N$ så $f(n) \leq g(n)$	☐	☐
Der findes $N \in \mathbb{N}$ så for alle $n \geq N$ er $f(n) \leq g(n)$	☐	☐
$g(n) = \Omega(f(n))$	☐	☐

(Opgavesættet fortsætter)

Opgave 4 (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 + i + i
```

```
Algoritme Loop2(n)
  i = 1
  while i ≤ n
    j = 1
    while j ≤ n
      j = j + j
      i = i + 1
```

```
Algoritme Loop3(n)
  for i = 1 to n
    j = 1
    while j ≤ i
      j = j + 1
```

Svar Loop1: _____

Svar Loop2: _____

Svar Loop3: _____

Opgave 5 (4 %)

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

```
Algoritme Loop1(n)
  i = 1
  j = 1
  while i ≤ n
    j = j + 1
    i = i + j
```

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

```
Algoritme Loop3(n)
  i = 2
  while i ≤ n
    i = i * i
    j = 1
    while j ≤ i
      j = j + 1
```

Svar Loop1: _____

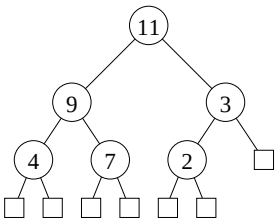
Svar Loop2: _____

Svar Loop3: _____

(Opgavesættet fortsætter)

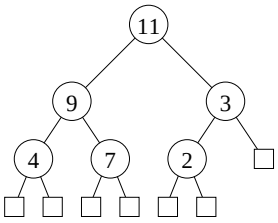
Opgave 6 (4 %)

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 7 (4 %)

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

Svar: _____

(Opgavesættet fortsætter)

Opgave 9 (4 %)

Betragt radix-sort anvendt på nedenstående liste af tal ($d = 4$, $k = 10$). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter de *to* mindst betydende cifre.

4227 1834 4400 0734 1327 9909

Svar: _____

Opgave 10 (4 %)

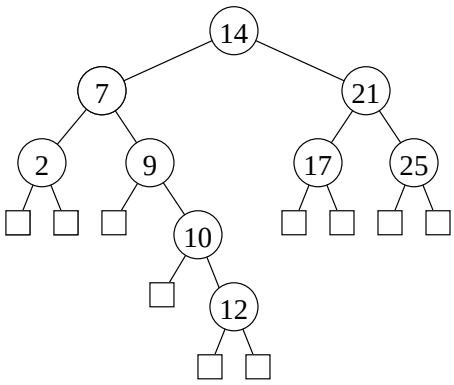
Angiv resultatet af at anvende PARTITION(A,3,13) på nedenstående array.

	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	15	3	18	5	9	11	24	12	14	10	7	22

Svar: _____

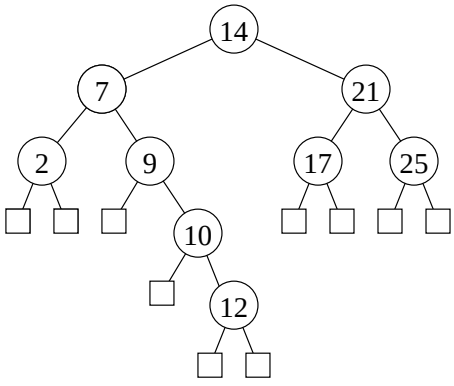
Opgave 11 (4 %)

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 11.



Svar: _____

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

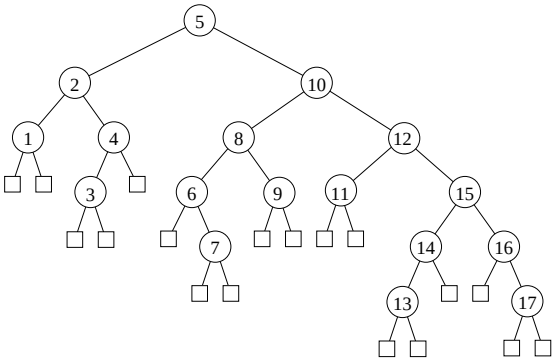


Svar: _____

(Opgavesættet fortsætter)

Opgave 12 (4 %)

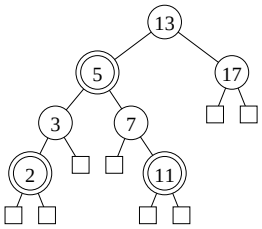
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.



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 %)

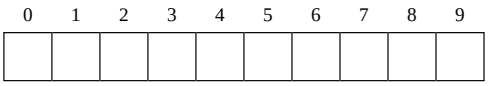
Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hash-funktionen er $h(k) = 3k \bmod 7$ og der indsættes elementerne 4, 1, 3, 9, 10, 7, og 2 i den givne rækkefølge.

Svar: _____

(Opgavesættet fortsætter)

Opgave 15 (4 %)

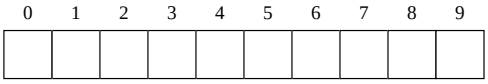
Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 6, 13, 0, 4, 8, 14, 7, og 3 indsættes i den givne rækkefølge, når hashfunktionen er $h(k) = 3k \bmod 10$.



Svar: _____

Opgave 16 (4 %)

Tegn hvordan en hashtabel der anvender *dobbelt hashing* ser ud efter at elementerne 2, 5, 7, 8, 3, 6, og 1 indsættes i den givne rækkefølge, når hashfunktionerne er $h_1(k) = 2k \bmod 10$ og $h_2(k) = 3k \bmod 10$, og hashtabellen har størrelse 10.



Svar: _____

Opgave 17 (4 %)

Angiv for hver af nedenstående datastrukturer indeholdende n elementer, hvor lang tid det tager at rapportere elementerne i datastrukturen i sorteret rækkefølge, som funktion af n i O-notation.

Rød-sort søgetræ ? Svar: _____

Binær max-heap ? Svar: _____

Hashtabel med linear probing ved 50% fyldningsgrad ? Svar: _____

(Opgavesættet fortsætter)

Opgave 18 (4 %)

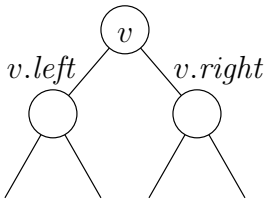
For en sorteret liste af tal $x_1, \ldots, x_n$, definerer vi

$$\text{sep}(x_1, \ldots, x_n) = \min\{x_n - x_{n-1}, x_{n-1} - x_{n-2}, \ldots, x_3 - x_2, x_2 - x_1\}$$

F.eks. er

$$\text{sep}(1, 3, 8, 11, 13, 19) = \min\{19 - 13, 13 - 11, 11 - 8, 8 - 3, 3 - 1\} = 2$$

Betragt et søgetræ hvor hver knude v ud over et element $v.e$, gemmer $v.sep$ som er lig sep(elementerne i v's undertræ), og $v.min$ og $v.max$ som er hhv. det mindste element og største element i v's undertræ. 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). F.eks. betegner $v.right.min$ det mindste element i v's højre undertræ.



Svar $v.min$ = _____

Svar $v.max$ = _____

Svar $v.sep$ = _____

Opgave 19 (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(a,c)
union(d,e)
union(b,d)
union(c,f)

Svar: _____

(Opgavesættet fortsætter)

Transitionssystem Halver-og-kvadrer
Konfigurationer: $\{[i,j] \mid \text{heltal } i,j \wedge i \geq 0 \wedge j \geq 0\}$

$[i,j] \rhd [i/2, j*j]$ **if** $i \geq 1 \wedge i$ lige
$[i,j] \rhd [i-1, j]$ **if** $i \geq 1$
$[i,j] \rhd [i, j-1]$ **if** $j \geq 1$

Opgave 20 (4 %)

For hvert af nedenstående udsagn, angiv om de er en invariant for ovenstående transitionssystem Halver-og-kvadrer. Startkonfigurationen antages at være $[n,2]$ hvor $n \geq 0$.

	Ja	Nej
$i \leq n$	☐	☐
$j \leq i$	☐	☐
$j \leq n$	☐	☐
$j^i \leq 2^n$	☐	☐
$i + j \leq n + 2$	☐	☐

Opgave 21 (4 %)

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

	Ja	Nej
$\mu(i,j) = i$	☐	☐
$\mu(i,j) = i + j$	☐	☐
$\mu(i,j) = j^i$	☐	☐
$\mu(i,j) = i + j^i$	☐	☐
$\mu(i,j) = n$	☐	☐

(Opgavesættet fortsætter)

Algoritme Loglog(n)
Inputbetingelse : heltal $n \geq 2$
Outputkrav : $r = \lfloor \log_2(\log_2(n)) \rfloor$
Metode : $p \leftarrow 2$
$r \leftarrow 0$
$\{I\}$ **while** $p * p \leq n$ **do**
$r \leftarrow r + 1$
$p \leftarrow p * p$

Opgave 22 (4 %)

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

	Ja	Nej
$p \geq r$	☐	☐
$p = 2^{2^r}$	☐	☐
$p = 2^r$	☐	☐
$n - r = p$	☐	☐
$p \leq n$	☐	☐

Opgave 23 (4 %)

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

	Ja	Nej
$\mu(r,p) = n - p$	☐	☐
$\mu(r,p) = p$	☐	☐
$\mu(r,p) = n - p^2$	☐	☐
$\mu(r,p) = n - r$	☐	☐
$\mu(r,p) = n - 2^{2^r}$	☐	☐

(Opgavesættet fortsætter)

Opgave 24 (4 %)

Givet et sorteret array A med n tal, $n \geq 1$, beregner nedenstående algoritme antal forskellige elementer i A. Denne værdi betegnes

$$\text{SetSize}(A) = 1 + |\{i \mid A[i-1] \neq A[i] \wedge 1 < i \leq n\}|$$

For at vise gyldigheden af algoritmen skal I_i og I_s være invarianter omkring variablerne i og s. Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke).

Algoritme ComputeSetSize(n)
Inputbetingelse : sorteret array A med n tal, $n \geq 1$
Outputkrav : $s = \text{SetSize}(A)$
Metode : $i \leftarrow 1$;
$s \leftarrow 1$;
$\{I_i \wedge I_s\}$ **while** $i < n$ **do**
if $A[i] \neq A[i+1]$ **do**
$s \leftarrow s + 1$;
$i \leftarrow i + 1$;

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: _____

Opgave 25 (4 %)

Antag en kø implementeres ved to stakke Front og Tail, og at operationerne på køen er implementeret som:

Algoritme Enqueue(x)
Metode : Tail.push(x)

Algoritme Dequeue()
Metode : **if** Front.empty() **then**
while not Tail.empty() **do**
Front.push(Tail.pop())
return Front.pop()

For at argumentere at operationerne tager amortiseret $O(1)$ tid kræves en potentiale funktion. Angiv en potentiale funktion hvorved tiden kan bevises. Argumentation for tiden kræves ikke.

Svar Φ(Front,Tail) : _____