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: Torsdag den 11. august 2011, kl. 9.00-11.00 |
| Eksamenslokale: Åbogade 34, 8200 Aarhus N, Benjamin bygningen |
| Tilladte medbragte hjælpemidler:<br>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

## Torsdag den 11. august 2011, 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 |
|---|---|---|
| $n^3 + n^3$ er $O(n^3)$ ? | ☐ | ☐ |
| $2^n$ er $O(n^2)$ ? | ☐ | ☐ |
| $\sqrt{n} + n^2$ er $O(n^{2.5})$ ? | ☐ | ☐ |
| $7n + 3n$ er $O(8n)$ ? | ☐ | ☐ |
| $n \cdot \log n$ er $O(n^2)$ ? | ☐ | ☐ |

**Opgave 2 (4 %)**

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

$$(\log n)^5$$
$$5n^2$$
$$\sqrt{n}$$
$$5^{\log n}$$
$$2^{\log n}$$

Svar: _____

**Opgave 3 (4 %)**

Betragt nedenstående sekvenser af $n$ elementer indsat i en binær max-heap i den givne rækkefølge ved hjælp af $n$ insert operationer. Det antages at $n$ er et lige tal. Angiv for hver af sekvenserne den samlede tid for indsættelserne som funktion af $n$ i $O$-notation.

$1, 2, 3, \ldots, n-1, n$ Svar: _____

$n, n-1, \ldots, 3, 2, 1$ Svar: _____

$1, 1, 1, \ldots, 1, 1$ Svar: _____

$n, n-1, n-2, n-3, \ldots, \frac{n}{2}+1, 1, 2, 3, \ldots, \frac{n}{2}$ Svar: _____

(Opgavesættet fortsætter)

## 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
    s = 0
    while s ≤ i
      s = s + 1
```

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

```
Algoritme Loop3(n)
  s = 1
  while s ≤ n
    s = 2 * s
```

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
  while i ≤ n
    j = 0
    while j ≤ n
      j = j + i
    i = 2 * i
```

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

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

Svar Loop1: _____

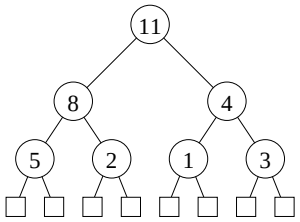
Svar Loop2: _____

Svar Loop3: _____

(Opgavesættet fortsætter)

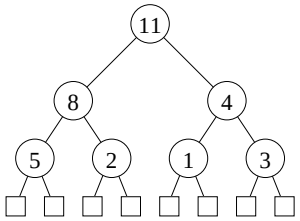
**Opgave 6 (4 %)**

Tegn hvordan nedenstående binære max-heap ser ud efter indsættelse af elementet 7.




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 3, 2, 1, 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 |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 8 | 5 | 2 | 10 | 7 | 3 | 6 | 9 | 1 |

Svar: _____

(Opgavesættet fortsætter)

**Opgave 9 (4 %)**

Betragt RADIX-SORT anvendt på nedenstående liste af tal ($d = 3$, $k = 10$). Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter det mindst betydende ciffer.

814 371 991 562 744 241 432

Svar: _____

**Opgave 10 (4 %)**

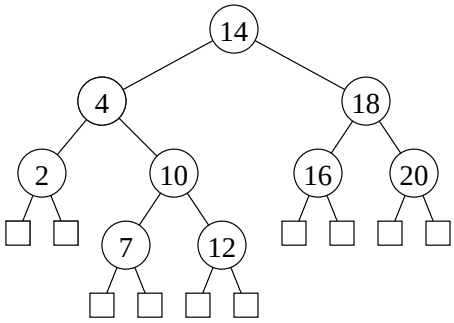
Angiv resultatet af at anvende PARTITION($A$,8,19) på nedenstående array.

| | 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 | 23 | 3 | 18 | 5 | 9 | 11 | 24 | 12 | 14 | 10 | 15 | 22 |

Svar: _____

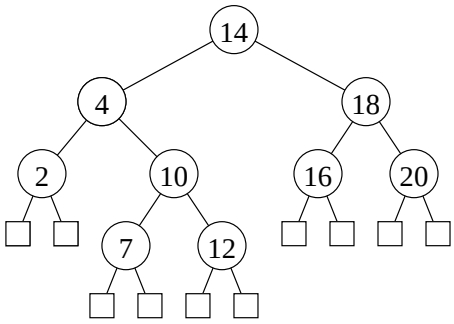
**Opgave 11 (4 %)**

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter indsættelse af elementet 17.




Svar: _____

Tegn hvordan nedenstående ubalancerede binære søgetræ ser ud efter slettelse af elementet 14.

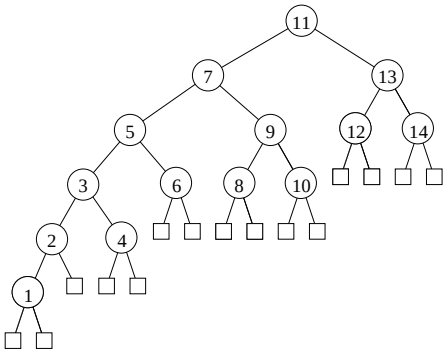



Svar: _____

(Opgavesættet fortsætter)

## Opgave 12 (4 %)

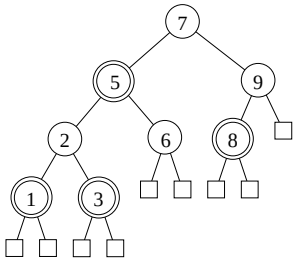
Angiv hvorledes knuderne i nedenstående binære søgetræ kan farves røde og sorte, således at det resulterende træ er et lovligt rød-sort træ.




Svar: _____

## Opgave 13 (4 %)

Tegn hvordan nedenstående rød-sorte træ (dobbeltcirkler angiver røde knuder) ser ud efter indsættelse af elementet 4.




Svar: _____

## Opgave 14 (4 %)

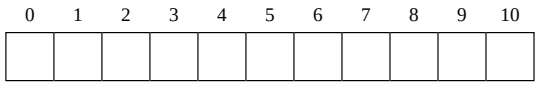
Tegn en hashtabel hvor der anvendes kædede lister til at håndtere kollisioner, når hash-funktionen er $h(k) = 4k \bmod 7$ og der indsættes elementerne 3, 2, 4, 8, 6, 11, 1 og 7 i den givne rækkefølge.

Svar: _____

(Opgavesættet fortsætter)

**Opgave 15 (4 %)**

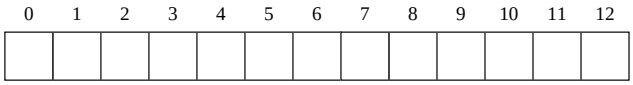
Tegn hvordan en hashtabel der anvender *linear probing* ser ud efter at elementerne 8, 2, 3, 1, 12, 16, 7 og 5 indsættes i den givne rækkefølge, når hashfunktionen er $h(k) = 2k \bmod 11$.




Svar: _____

**Opgave 16 (4 %)**

Tegn hvordan en hashtabel der anvender *dobbelt hashing* ser ud efter at elementerne 17, 20, 4, 8, 7, 12 og 6 indsættes i den givne rækkefølge, når hashfunktionerne er $h_1(k) = k \bmod 13$ og $h_2(k) = 1 + (2k \bmod 7)$, og hashtabellen har størrelse 13.




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

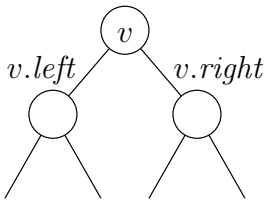
Svar: _____

## Opgave 18 (4 %)

Betragt et søgetræ hvor hver knude $v$ ud over et element $v.e$ gemmer en værdi $v.x$ (søgetræet er ordnet efter $v.e$). For at anvende træet til statistiske formål udvider vi hver knude $v$ med følgende information:

$v.size$ = antal knuder i $v$'s undertræ $T_v$
$v.sum = \sum_{u \in T_v} u.x$
$v.sumofsquares = \sum_{u \in T_v} (u.x)^2$

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 $v.size$ = _____

Svar $v.sum$ = _____

Svar $v.sumofsquares$ = _____

## Opgave 19 (4 %)

Angiv worst-case tiden for at finde det $\lceil \sqrt{n} \rceil$ mindste element blandt $n$ elementer, når man *i*) konstruerer en binær min-heap med Build-Min-Heap og kalder Heap-Extract-Min $\lceil \sqrt{n} \rceil$ gang, og *ii*) kalder Randomized-Select.

Svar min-heap: _____

Svar Randomized-Select: _____

(Opgavesættet fortsætter)

**Transitionssystem** Nedtælling
Konfigurationer: $\{[i, j, s] \mid$ ikke negative heltal $i, j, s\}$

$[i, j, s] \rhd [i, j - 1, s + 1]$ **if** $j > 0$
$[i, j, s] \rhd [i - 1, i - 1, s + 1]$ **if** $j = 0 \wedge i > 0$

## Opgave 20 (4 %)

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

| | Ja | Nej |
|---|---|---|
| $i^2 + i + 2j + 2s = n^2 + n$ | ☐ | ☐ |
| $i + j = n$ | ☐ | ☐ |
| $i^2 \leq n^2 - j$ | ☐ | ☐ |
| $i \cdot n = j$ | ☐ | ☐ |
| $i \geq j$ | ☐ | ☐ |

## Opgave 21 (4 %)

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

| | Ja | Nej |
|---|---|---|
| $\mu(i, k) = i$ | ☐ | ☐ |
| $\mu(i, k) = s$ | ☐ | ☐ |
| $\mu(i, k) = i^2 + i + 2j + 2s$ | ☐ | ☐ |
| $\mu(i, k) = i + j$ | ☐ | ☐ |
| $\mu(i, k) = i^2 + i + 2j$ | ☐ | ☐ |

(Opgavesættet fortsætter)

**Algoritme** ZeroCount($n$)
Inputbetingelse : Array $A[1..n]$, hvor $A[k] \in \{0,1\}$ for $1 \leq k \leq n$
Outputkrav : $i = |\{k \mid 1 \leq k \leq n \wedge A[k] = 0\}|$
Metode : $i \leftarrow 0$
$j \leftarrow n$
$\{I\}$ **while** $i < j$ **do**
**if** $A[j] = 1$ **then**
$j \leftarrow j - 1$
**else**
$i \leftarrow i + 1$
$A[j] \leftarrow A[i]$
$A[i] \leftarrow 0$

## Opgave 22 (4 %)

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

| | Ja | Nej |
|---|---|---|
| $0 \leq i < j \leq n$ | ☐ | ☐ |
| $i = \|\{k \mid 1 \leq k \leq j \wedge A[k] = 0\}\|$ | ☐ | ☐ |
| $A[k] = 0$ for $1 \leq k \leq i$ | ☐ | ☐ |
| $A[k] = 1$ for $j \leq k \leq n$ | ☐ | ☐ |
| $i + j = n$ | ☐ | ☐ |

## Opgave 23 (4 %)

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

| | Ja | Nej |
|---|---|---|
| $\mu(j,r) = j$ | ☐ | ☐ |
| $\mu(j,r) = j - i$ | ☐ | ☐ |
| $\mu(j,r) = (n-i)j$ | ☐ | ☐ |
| $\mu(j,r) = n - ij$ | ☐ | ☐ |
| $\mu(j,r) = n - j$ | ☐ | ☐ |

(Opgavesættet fortsætter)

**Opgave 24 (4 %)**

Givet et array $A[1..n]$ af heltal definerer vi dominate($i$) som udsagnet:

$$\text{dominate}(i) \text{ er sand} \Leftrightarrow A[i] \geq A[k] \text{ for alle } 1 \leq k \leq i$$

Nedenstående algoritme beregner antallet af elementer i $A$ hvor dominate($i$) er sand. For at vise gyldigheden af algoritmen skal $I_m$ og $I_d$ være invarianter omkring variablerne $m$ og $d$. Angiv invarianter hvormed gyldigheden af algoritmen kan bevises (bevis for invarianterne kræves ikke).

**Algoritme** Dominate($n$)
Inputbetingelse : array $A[1..n]$ af positive heltal, $n \geq 1$
Outputkrav : $d = |\{i \mid 1 \leq i \leq n \wedge \text{dominate}(i)\}|$
Metode : $i \leftarrow 2$
$d \leftarrow 1$
$m \leftarrow A[1]$
$\{I_d \ \wedge \ I_m\}$ **while** $i \leq n$ **do**
**if** $A[i] \geq m$ **then**
$m \leftarrow A[i]$
$d \leftarrow d + 1$
$i \leftarrow i + 1$

Svar $I_m$: _____

Svar $I_d$: _____

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 $\mu$: _____

**Opgave 25 (4 %)**

Antag at vi har en sorteret kædet liste $L$ og vi har en operation Insert-and-Cut-off($x$), som indsætter $x$ i $L$ efter at have fjernet alle elementer fra $L$ der er mindre end $x$, dvs. vi sletter alle elementer fra fronten af listen indtil vi finder det første element $\geq x$.

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

Svar $\Phi$ = _____