INSTITUT FOR DATALOGI, AARHUS UNIVERSITET

Science and Technology
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 1 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 12
Eksamensdag: Tirsdag den 20. juni 2017, kl. 9.00-11.00
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)

Institut for Datalogi
Aarhus Universitet

Tirsdag den 20. juni 2017, kl. 9.00-11.00

Dette eksamenssæt består af en mængde multiple-choice-opgaver. Opgaverne besvares på opgaveformuleringen **som afleveres**.

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

Hvert delspørgsmål har præcist et rigtigt svar. For hvert delspørgsmål, kan du vælge **max ét svar** ved at afkrydse den tilsvarende rubrik. Et 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 opgave med vægt $v\,\%$ og med n delspørgsmål, hvor du opnår samlet s point, beregnes din besvarelse af opgaven som:

$$\frac{s}{n} \cdot v\,\%$$

Bemærk at det er muligt at få negative point for en opgave.

Opgave 1 (10 %)

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej
n^6 er $O(n^9)$?	☐	☐
$n^{1/6}$ er $O(n^{1/9})$?	☐	☐
$9n$ er $O(6n)$?	☐	☐
$9n$ er $O(6+n)$?	☐	☐
n er $O(6^9)$?	☐	☐
n^6+n^9 er $O(n^6)$?	☐	☐
$1/n$ er $O(2^{1/n})$?	☐	☐
$1/n$ er $O(1/n^2)$?	☐	☐
$6\log n$ er $O(\log(9n))$?	☐	☐
n^2 er $O(3^n)$?	☐	☐
n^3 er $O(2^n)$?	☐	☐
$(\log n)^3$ er $O(2\log n)$?	☐	☐
$3^{\log n}$ er $O(n)$?	☐	☐
$(\log n)(\log n)$ er $O(2\log n)$?	☐	☐
$\sqrt{n}/\log n$ er $O(n^{1/3})$?	☐	☐
6^9 er $O(9^6)$?	☐	☐
n^4 er $O(n\cdot n+n\cdot n)$?	☐	☐
$(\log n)^2$ er $O(\sqrt{n}))$?	☐	☐
$n^2\cdot n^3$ er $O(n^4)$?	☐	☐
n^2+n^3 er $O(n^4)$?	☐	☐

Opgave 2 (4 %)

For en binær max-heap af størrelse n, angiv best-case og worst-case udførselstid for nedenstående operationer.

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n\log n)$
INSERT, worst-case	☐	☐	☐	☐
INSERT, best-case	☐	☐	☐	☐
HEAP-EXTRACT-MAX, worst-case	☐	☐	☐	☐
HEAP-EXTRACT-MAX, best-case	☐	☐	☐	☐
BUILD-MAX-HEAP, worst-case	☐	☐	☐	☐
BUILD-MAX-HEAP, best-case	☐	☐	☐	☐

Opgave 3 (10 %)

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 n
      $s = s + 1$

Algoritme Loop2(n)
  $s = 0$
  for $i = 1$ to n
    for $j = 1$ to n
      if $i = j$ **then**
        for $k = 1$ to n
          $s = s + 1$

Algoritme Loop3(n)
  $i = 1$
  $j = 1$
  while $i \leq n$
    while $j \leq i$
      $j = j + 1$
    $i = 2 * i$

Algoritme Loop4(n)
  $i = 1$
  $j = 1$
  $s = 1$
  while $i \leq n$
    if $i = j$ **then**
      for $k = 1$ to n
        $s = s + 1$
      $j = 2 * j$
    $i = i + 1$

Algoritme Loop5(n)
  $i = 1$
  $j = n$
  while $i \leq j$
    $i = i + 1$
    $j = j - 1$

Algoritme Loop6(n)
  $i = 1$
  $j = n$
  while $i \leq j$
    $i = 4 * i$
    $j = 2 * j$

	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n\sqrt{n})$	$O(\sqrt{n})$	$O((\log n)^2)$	$O(n^3)$
Loop1	☐	☐	☐	☐	☐	☐	☐	☐
Loop2	☐	☐	☐	☐	☐	☐	☐	☐
Loop3	☐	☐	☐	☐	☐	☐	☐	☐
Loop4	☐	☐	☐	☐	☐	☐	☐	☐
Loop5	☐	☐	☐	☐	☐	☐	☐	☐
Loop6	☐	☐	☐	☐	☐	☐	☐	☐

Opgave 4 (4 %)

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

Angiv hvordan ovenstående binære max-heap ser ud efter Heap-Extract-Max.

1	2	3	4	5	6	7	8	9	
13	10	7	5	9	12	4	3	2	☐

1	2	3	4	5	6	7	8	9	
13	10	12	5	9	4	3	2	7	☐

1	2	3	4	5	6	7	8	9	
13	10	12	5	9	7	4	3	2	☐

1	2	3	4	5	6	7	8	9	
13	12	10	9	7	5	4	3	2	☐

Opgave 5 (4 %)

Angiv den binære max-heap efter indsættelse af elementerne 3, 2, 4, 6, 5, 7, og 8 i den givne rækkefølge, startende med den tomme heap.

1	2	3	4	5	6	7	
8	7	6	5	4	3	2	☐

1	2	3	4	5	6	7	
8	5	7	2	4	3	6	☐

1	2	3	4	5	6	7	
3	2	4	6	5	7	8	☐

1	2	3	4	5	6	7	
8	6	7	2	5	3	4	☐

Opgave 6 (4 %)

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

Angiv hvordan ovenstående array ser ud efter anvendelsen af BUILD-MAX-HEAP.

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

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

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

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

Opgave 7 (4 %)

Betragt RADIX-SORT anvendt på nedenstående liste af tal ($d = 4$, $k = 5$).

4231 4321 1432 2431 3421

Angiv den delvist sorterede liste efter at radix-sort har sorteret tallene efter de *to* mindst betydende cifre.

3421	4321	4231	2431	1432	☐
3421	4321	2431	4231	1432	☐
4321	3421	2431	4231	1432	☐
4321	3421	4231	2431	1432	☐
1432	2431	3421	4231	4321	☐

Opgave 8 (4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	5	19	7	26	10	8	23	6	12	17	30	42	37

Angiv resultatet af at anvende PARTITION(A,3,11) på ovenstående array.

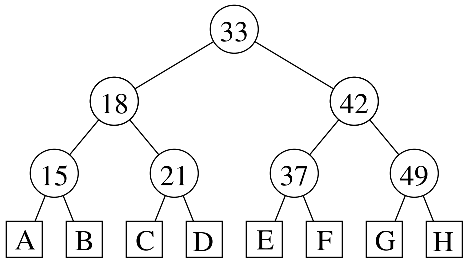
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
2	5	7	10	8	6	19	23	26	12	17	30	42	37	☐

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
2	5	7	10	8	6	12	17	19	23	26	30	42	37	☐

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
2	5	7	10	8	6	12	17	19	26	23	30	42	37	☐

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
2	5	6	7	8	10	12	17	19	23	26	30	37	42	☐

Opgave 9 (4 %)

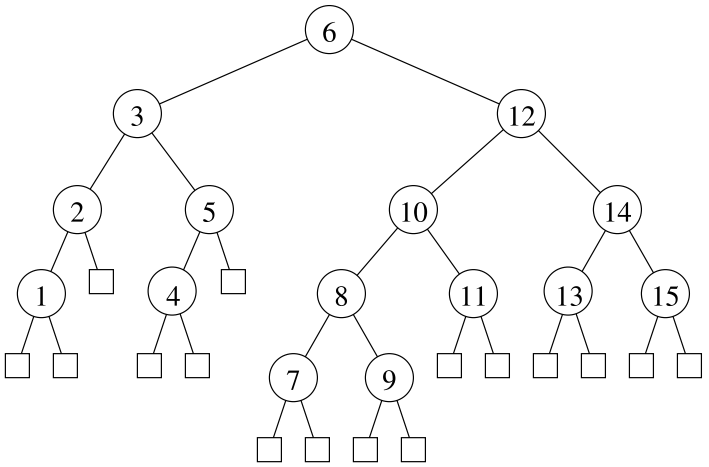


Angiv i hvilke blade A-H i ovenstående ubalancerede binære søgetræ elementerne 6, 22, 41, 19, og 34 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående syv elementer).

	A	B	C	D	E	F	G	H
`Insert`(6)	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(22)	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(41)	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(19)	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(34)	☐	☐	☐	☐	☐	☐	☐	☐

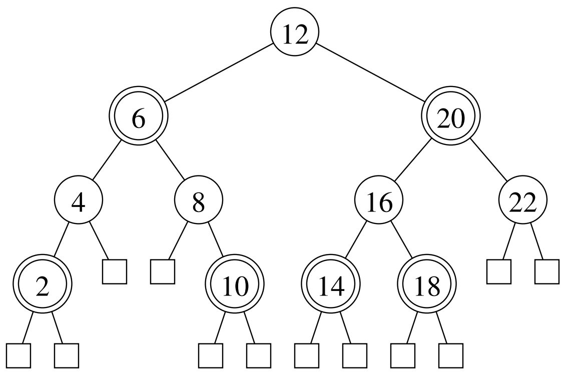
Opgave 10 (4 %)

For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde



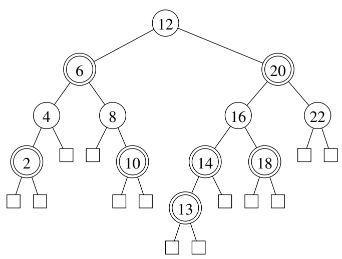
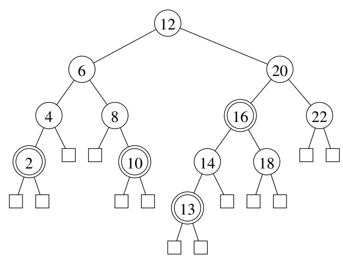
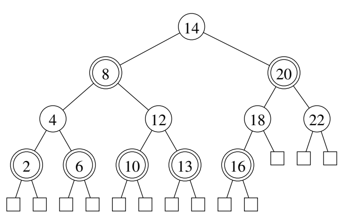
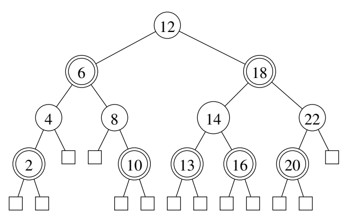
	Ja	Nej
$1, 4, 8, 12$	☐	☐
$2, 5, 7, 9, 12$	☐	☐
$1, 4, 7, 9, 12$	☐	☐
$1, 4, 7, 9, 10, 14$	☐	☐
$1, 4, 7, 9, 10, 13, 15$	☐	☐

Opgave 11 (4 %)



Angiv det resulterende rød-sorte træ når man indsætter 13 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

☐ ☐ ☐ ☐

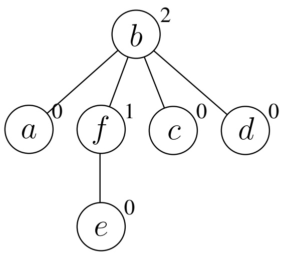





Opgave 12 (4 %)

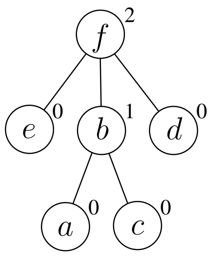
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

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

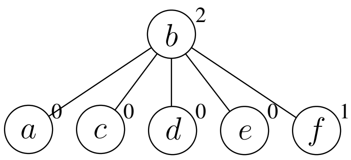
☐



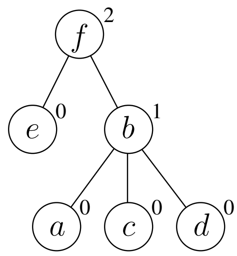
☐



☐



☐



Opgave 13 (4 %)

I følgende hashtabel er anvendt *linear probing* med hashfunktionen $h(k) = (2k) \bmod 11$.

0	1	2	3	4	5	6	7	8	9	10
22			18	2			9			16

Angiv positionerne de tre elementer 5, 7 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 2, 9, 16, 18, 22).

	0	1	2	3	4	5	6	7	8	9	10
`Insert`(5)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(7)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(11)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Opgave 14 (4 %)

I følgende hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = k \bmod 11$ og $h_2(k) = 1 + (3k \bmod 10)$.

0	1	2	3	4	5	6	7	8	9	10
			14	15		17	7			10

Angiv positionerne de tre elementer 3, 4 og 6 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 7, 10, 14, 15 og 17).

	0	1	2	3	4	5	6	7	8	9	10
`Insert`(3)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(4)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
`Insert`(6)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Opgave 15 (4 %)

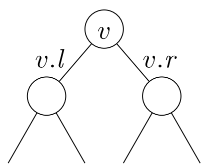
For n tal $x_1, \ldots, x_n$ ønsker vi at beregne koefficienterne a og b til polynomiet

$$P(y) = \sum_{i=1}^{n} (x_i - y)^2 = n \cdot y^2 + a \cdot y + b \ .$$

F.eks. for $x_1 = 2$, $x_2 = 3$, og $x_3 = 5$, har vi polynomiet $P(y) = (2 - y)^2 + (3 - y)^2 + (5 - y)^2 = 3y^2 - 20y + 38$, dvs. $a = -20$ og $b = 38$.

Betragt et søgetræ hvor hver knude v gemmer et tal $v.x$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v to værdier $v.a$ og $v.b$, som er a og b koeffienterne for $P(y)$ polynomiet defineret ved alle x værdierne i v's undertræ.

Angiv hvorledes $v.a$ og $v.b$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).



$$v.a = \begin{cases} v.l.a + v.r.a + v.x & \square \\ v.l.a + v.r.a - 2 * v.x & \square \\ v.l.a * v.r.a * (v.x)^2 & \square \\ v.l.a + v.r.a + (v.x)^2 & \square \end{cases}$$

$$v.b = \begin{cases} v.l.b + v.r.b + v.x & \square \\ v.l.b + v.r.b + (v.x)^2 & \square \\ v.l.b + v.r.b - 2 * v.x & \square \\ v.l.b + v.r.b + 2 * v.x & \square \end{cases}$$

Transitionssystem CHASE
Konfigurationer: $\{[x, y] \mid \text{heltal } x, y \geq 0\}$
$[x, y] \triangleright [x + 2, y + 1] \quad \textbf{if} \quad x < y$

Opgave 16 (4 %)

For hvert af nedenstående udsagn, angiv om de er en invariant for ovenstående transitionssystem CHASE. Startkonfigurationen antages at være $[x_0, y_0]$, hvor $x_0 \geq 0$ og $y_0 \geq 0$.

	Ja	Nej
$2(x - x_0) = y - y_0$	☐	☐
$x - x_0 = 2(y - y_0)$	☐	☐
$x \leq y$	☐	☐
$\lvert y - x\rvert \leq \lvert y_0 - x_0\rvert$	☐	☐
$\lvert x\rvert \leq \lvert y\rvert$	☐	☐

Opgave 17 (4 %)

For hver af nedenstående funktioner, angiv om de er en termineringsfunktion for ovenstående transitionssystem CHASE. Startkonfigurationen antages at være $[x_0, y_0]$, hvor $x_0 \geq 0$ og $y_0 \geq 0$.

	Ja	Nej
$\mu(x, y) = x - x_0$	☐	☐
$\mu(x, y) = x_0 - x$	☐	☐
$\mu(x, y) = y - x$	☐	☐
$\mu(x, y) = \lvert y - x\rvert$	☐	☐
$\mu(x, y) = 2y - x$	☐	☐

Opgave 18 (4 %)

For QuickSort på input af størrelse n, og et givet element e i inputet, hvor mange sammenligninger vil dette element e indgå i under udførslen af QuickSort? Forventet antal sammenligninger er her forventet antal sammenligninger for en tilfældig permutation af input.

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$
Worst-case antal sammenligninger	☐	☐	☐	☐	☐
Best-case antal sammenligninger	☐	☐	☐	☐	☐
Forventet antal sammenligninger	☐	☐	☐	☐	☐

Givet et heltal $n \geq 0$ heltal, så identificerer nedenstående algoritme antallet af 1-tal i den binære repræsentation af n, dvs. beregner

$$bits(n) = |\{i \mid 0 \leq 2^i \leq n \ \wedge \ n \bmod 2^{i+1} \geq 2^i\}|.$$

Algoritme BITS(n)
Inputbetingelse : Heltal $n \geq 0$
Outputkrav : $r = bits(n)$
Metode : $x \leftarrow n$;
$r \leftarrow 0$;
$\{I\}$ **while** $x > 0$ **do**
  if $x \bmod 2 = 1$ **then**
    $r \leftarrow r + 1$
    $x \leftarrow x - 1$
  else
    $x \leftarrow x/2$

Opgave 19 (4 %)

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

	Ja	Nej
$x > 0$	☐	☐
$0 \leq x \leq n$	☐	☐
$r = bits(x)$	☐	☐
$r = bits(n - x)$	☐	☐
$r = bits(n) - bits(x)$	☐	☐

Opgave 20 (4 %)

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

	Ja	Nej
$\mu(x, r, n) = x$	☐	☐
$\mu(x, r, n) = bits(x)$	☐	☐
$\mu(x, r, n) = r$	☐	☐
$\mu(x, r, n) = x + r$	☐	☐
$\mu(x, r, n) = 2x + r$	☐	☐

Opgave 21 (4 %)

Givet et array $C[1..n]$ med $n \geq 1$ heltal, og et heltal x, så beregner nedenstående algoritme

$$Poly(C,x) = \sum_{i=1}^{n} C[i] \cdot x^{i-1} = C[1] + C[2] \cdot x + C[3] \cdot x^2 + \cdots + C[n] \cdot x^{n-1} \ .$$

Algoritme POLY(C, x)
Inputbetingelse : Heltal x og array C med $n \geq 1$ heltal
Outputkrav : $r = Poly(C, x)$
Metode : $i \leftarrow n$;
$r = C[n]$;
$\{I\}$ **while** $i > 1$ **do**
  $i \leftarrow i - 1$;
  $r \leftarrow r * x + C[i]$

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

	Ja	Nej
$r = Poly(C[1..i], x)$	☐	☐
$r = Poly(C[i..n], x)$	☐	☐
$r = Poly(C[1..n], x)$	☐	☐
$1 < i < n$	☐	☐
$1 \leq i \leq n$	☐	☐

Opgave 22 (4 %)

Antag at et array X af størrelse n indeholder to stakke S og T, henholdsvis af størrelse s og t, således at $X[1..s] = S$ og $X[n+1-t..n] = T$, hvor toppen af de to stakke er henholdsvis $X[s]$ og $X[n+1-t]$. Når X bliver fuld, dvs. $s+t = n$, allokeres et dobbelt så stort array til X, og S og T kopieres over i dette array.

Med en passende potentialefunktion kan man argumentere for at stakoperationerne Push og Pop på de to stakke tager amortiseret $O(1)$ tid. Angiv for hver af nedenstående funktioner om dette er en sådan potentialefunktion Φ.

	Ja	Nej
$s + t$	☐	☐
$t - s$	☐	☐
$s + n - t$	☐	☐
$\max\{0, 2(s+t) - n\}$	☐	☐
$n - s - t$	☐	☐