EKSAMEN **Grundlæggende Algoritmer og Datastrukturer** **Torsdag den 24. januar 2019, kl. 9.00–11.00**
Institut for Datalogi, Science and Technology, Aarhus Universitet
Antal sider i opgavesættet (incl. forsiden): 13
Tilladte medbragte hjælpemidler: **Ingen**

Studienummer : ____________________

Navn : ____

Vejledning og pointgivning

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 må 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 (6 %)

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

	Ja	Nej
n^7 er $O(7n)$?	A	B
2^n er $O(n^n)$?	A	B
$\sqrt{\log n}$ er $O(\log n)$?	A	B
$n^2 + n^5$ er $O(n^6)$?	A	B
$5 \cdot 5^5$ er $O(4)$?	A	B
$\log n$ er $O(n^{0.001})$?	A	B
$3^{\log n}$ er $O(n^2)$?	A	B
$3n^7 + 4n^2$ er $O(10n^6)$?	A	B
$3^{\log n}$ er $O((\log n)^3)$?	A	B
$\sqrt{n} \cdot \log n$ er $O(\sqrt{n \log n})$?	A	B
3^n er $O(n^3)$?	A	B
$\sum_{i=1..n} i$ er $O(n \log n)$?	A	B

Opgave 2 (4 %)

Angiv worst-case udførselstiden for hver af nedenstående algoritmer når input er et array af størrelse n.

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^3)$
BinarySearch	A	B	C	D	E	F	G
InsertionSort	A	B	C	D	E	F	G
MergeSort	A	B	C	D	E	F	G
HeapSort	A	B	C	D	E	F	G
QuickSort	A	B	C	D	E	F	G
Partition	A	B	C	D	E	F	G

Opgave 3 (4 %)

Angiv hvor mange gange det *største element* i et array med n elementer i worst-case kan blive sammenlignet med andre elementer under udførslen af MergeSort.

$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$
A	B	C	D	E

Opgave 4 (6 %)

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

Algoritme Loop1(n)
  $i = 1$
  while $i \leq n$
    $i = i + 1$

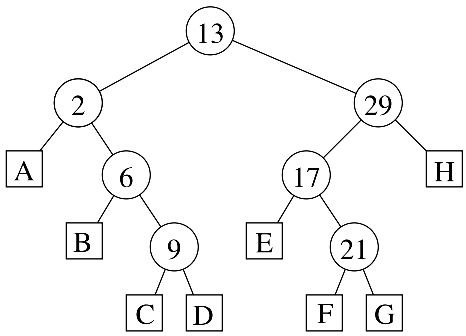
Algoritme Loop2(n)
  $i = 1$
  while $i \leq n * n$
    $i = 2 * i$

Algoritme Loop3(n)
  $i = 1$
  while $i \leq n$
    $j = 0$
    while $j \leq n$
      $j = j + 1$
    $i = 2 * i$

Algoritme Loop4(n)
  $i = 1$
  $s = 0$
  while $s \leq n$
    $s = s + i$
    $i = i + 1$

	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta((\log n)^2)$	$\Theta(n^3)$
Loop1	A	B	C	D	E	F	G	H
Loop2	A	B	C	D	E	F	G	H
Loop3	A	B	C	D	E	F	G	H
Loop4	A	B	C	D	E	F	G	H

Opgave 5 (4 %)



Angiv i hvilke blade A-H i ovenstående ubalancerede binære søgetræ elementerne 7, 18, -5, 14, og 23 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående syv elementer).

	A	B	C	D	E	F	G	H
`Insert`(7)	A	B	C	D	E	F	G	H
`Insert`(18)	A	B	C	D	E	F	G	H
`Insert`(-5)	A	B	C	D	E	F	G	H
`Insert`(14)	A	B	C	D	E	F	G	H
`Insert`(23)	A	B	C	D	E	F	G	H

Opgave 6 (4 %)

I følgende hashtabel er anvendt *linear probing* med hashfunktionen $h(k) = (5k + 4) \bmod 11$.

0	1	2	3	4	5	6	7	8	9	10
8			2	11	9				12	1

Angiv positionerne de tre elementer 0, 5 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 2, 8, 9, 11 og 12).

	0	1	2	3	4	5	6	7	8	9	10
`Insert`(0)	A	B	C	D	E	F	G	H	I	J	K
`Insert`(5)	A	B	C	D	E	F	G	H	I	J	K
`Insert`(10)	A	B	C	D	E	F	G	H	I	J	K

Opgave 7 (4 %)

Angiv den binære max-heap efter indsættelse af elementerne 1, 4, 6, 3, 5, 2 og 7 i den givne rækkefølge, startende med den tomme heap.

1	2	3	4	5	6	7	
7	5	6	1	2	3	4	A
7	3	6	1	5	2	4	B
7	5	6	1	3	2	4	C
7	6	5	4	3	2	1	D

Opgave 8 (4 %)

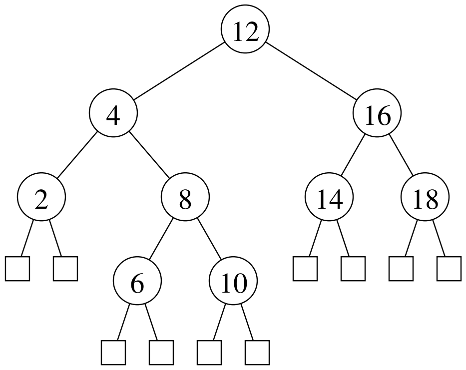
1	2	3	4	5	6	7
7	6	5	1	4	2	3

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

1	2	3	4	5	6	
3	6	5	1	4	2	A
6	4	5	1	3	2	B
6	5	1	4	2	3	C
6	4	5	1	2	3	D

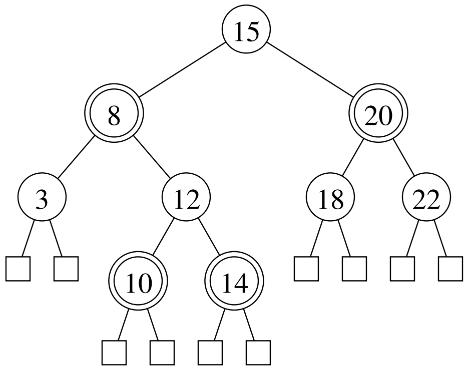
Opgave 9 (4 %)

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



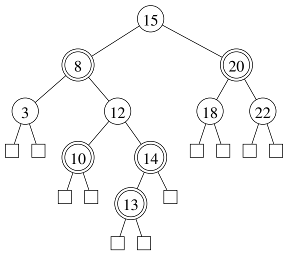
	Ja	Nej
$\emptyset$	A	B
$4, 16$	A	B
$4, 8, 16$	A	B
$8, 12$	A	B
$4, 6, 10, 14, 18$	A	B

Opgave 10 (4 %)

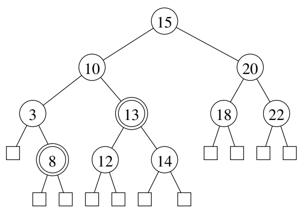


Angiv det resulterende rød-sorte træ når man indsætter 13 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

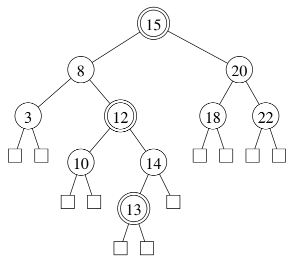
A



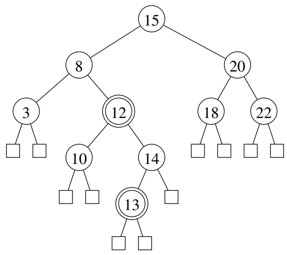
B



C



D

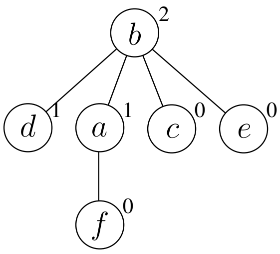


Opgave 11 (4 %)

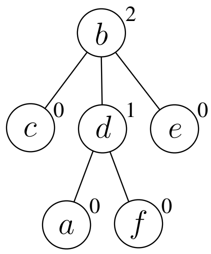
Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

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

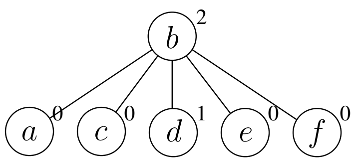
A



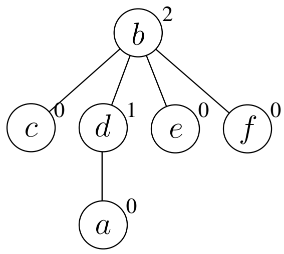
B



C



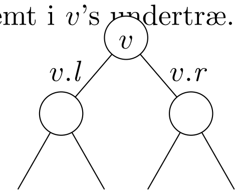
D



Opgave 12 (4 %)

Et interval træ er et rød-sort træ hvor hver knude gemmer præcis ét interval ($low, high$), og intervallerne er sorteret fra venstre-mod-højre efter stigende low værdi. For en knude v i træet ladet vi $v.low$ og $v.high$ betegne endepunkterne på intervallet gemt i knuden. Desuden gemmer v værdien $v.max$ som er den maximale værdi i et interval gemt i v's undertræ.

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



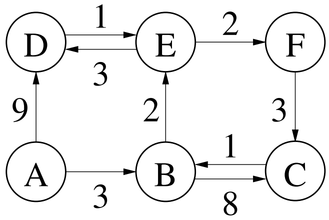
$$v.max = \begin{cases} v.r.max & \boxed{\text{A}} \\ \max(v.r.max, v.high) & \boxed{\text{B}} \\ \max(v.r.max, v.l.max, v.high) & \boxed{\text{C}} \\ \max(v.l.high, v.high, v.r.high) & \boxed{\text{D}} \end{cases}$$

Opgave 13 (4 %)

Hver af følgende rekursionsligninger har basistilfældet $T(1) = 1$. Angiv for hver ligning, hvad løsningen er.

	$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 1 + T(n/2)$	A	B	C	D	E	F	G	H
$T(n) = 2 \cdot T(n/2) + n$	A	B	C	D	E	F	G	H
$T(n) = T(n/2) + n$	A	B	C	D	E	F	G	H
$T(n) = 4 \cdot T(n/2) + n^2$	A	B	C	D	E	F	G	H

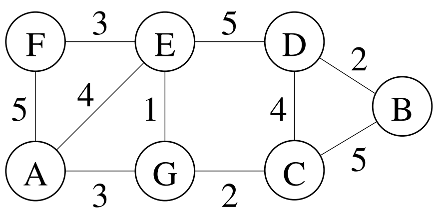
Opgave 14 (4 %)



Antag Dijkstras algoritme anvendes til at finde korteste **afstande fra A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A B D C E F	A B E F C D	A B E D F C	A B E F D C
A	B	C	D

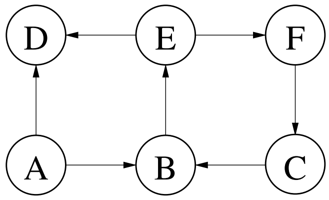
Opgave 15 (4 %)



Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen **starter i knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A G E F C D B	A G E C F D B	A E G C D B F	A E G C B D F
A	B	C	D

Opgave 16 (4 %)



Betragt et DFS-gennemløb af ovenstående graf, hvor DFS-gennemløbet **starter i knuden** A, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge.

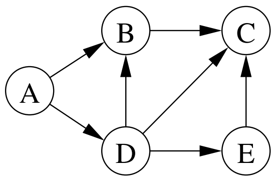
Angiv i hvilken rækkefølge knuderne får tildelt "discovery time".

A	B	C	D
A D B E F C	A B E D F C	A B E F C D	A B E F D C

Angiv for hver af nedenstående kanter kvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge
(A, D)	A	B	C	D
(B, E)	A	B	C	D
(C, B)	A	B	C	D

Opgave 17 (4 %)



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej
A B C D E	A	B
A B D E C	A	B
A D B E C	A	B
A D E B C	A	B
A B C E D	A	B

Opgave 18 (4 %)

Givet et positivt heltal n, så beregner nedenstående algoritme n^2.

Algoritme Square(n)
Inputbetingelse : Heltal $n \geq 1$
Outputkrav : $r = n^2$
Metode : $r \leftarrow 0$
$i \leftarrow 0$
$\{I\}$ **while** $i < n$ **do**
  $i \leftarrow i + 1$
  $r \leftarrow r + i$
$r \leftarrow 2 * r - n$

For hvert af nedenstående udsagn, angiv om de er en invariant I for algoritmen Square.

	Ja	Nej
$i \geq 0 \ \wedge \ r = i^2$	A	B
$0 \leq i \leq n$	A	B
$i \geq 0 \ \wedge \ r = i(i+1)/2$	A	B
$r \geq i \geq 0$	A	B
$0 \leq r \leq n$	A	B

Opgave 19 (4 %)

Selektionsalgoritmen til at finde det $i$te mindste element i et ikke-sorteret list i worst-case tid $O(n)$ [CLRS, kapitel 9.3], deler input op i grupper af 5 elementer, finder rekursivt medianen af gruppernes medianer, bruger det fundne element som pivot til en opdeling, og kalder rekursivt på en af de to dele. Udførselstiden kan beskrives ved følgende rekursionsligning (afrunding ignoreres):

$$T(n) = T(n/5) + T(7n/10) + c \cdot n$$

Hvad bliver udførselstiden hvis man ændrer grupperne i algoritmen til at have størrelse 3 istedet for 5?

$\Theta(\log n)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$
A	B	C	D

Opgave 20 (4 %)

Antag en liste L bruges til at opbevare en mængde af forskellige tal. Følgende tre operationer ønskes understøttet: Add(x) tilføjer x til listen (x kan antages at være forskellig fra alle tal i listen), RemoveSmallerHalf og RemoveLargerHalf fjerner henholdsvis de $\lfloor |L|/2 \rfloor$ mindste og $\lceil |L|/2 \rceil$ største elementer fra listen. Add(x) tilføjer blot det nye element bagerst i listen i worst-case $O(1)$ tid. RemoveSmallerHalf og RemoveLargerHalf anvender først deterministisk selektion til først at finde det $\lfloor |L|/2 \rfloor + 1$ mindste element e i L i worst-case $O(|L|)$ tid, hvorefter L løbes igennem og alle elementer fjernes der er henholdsvis mindre end e eller større end eller lig e.

Med en passende potentialefunktion kan man argumentere for at alle tre operationer tager amortiseret $O(1)$ tid. Angiv for hver af nedenstående om dette er en sådan potentialefunktion Φ.

	Ja	Nej
$\lvert L \rvert$	A	B
$\log \lvert L \rvert$	A	B
$2\lvert L \rvert$	A	B
$\lvert L \rvert^2$	A	B
$\lvert L \rvert \cdot \log \lvert L \rvert$	A	B

Dynamisk programmering

De næste fire opgaver vedrører at løse *subset sum* problemet ved hjælp af dynamisk programmering.

Givet en liste af n positive tal $L = (x_1, x_2, \ldots, x_n)$ og et heltal V, ønsker vi at afgøre om der findes en delsekvens af tallene som har sum V. F.eks. for $L = (3, 10, 17, 25)$ og $V = 45$ er svaret ja, da $45 = 3 + 17 + 25$, hvorimod for $V = 46$ findes ingen delsekvenser af L med sum 46.

For $0 \le k \le n$ og $0 \le v \le V$ lader vi $S(k, v)$ angive sandhedsværdien om værdien v kan opnås som en delsekvens af de k første elementer i L, dvs. om der findes en delsekvens af $(x_1, x_2, \ldots, x_k)$ med sum v.

$S(k, v)$ kan bestemmes ved følgende rekursionsformel.

$$S(k,v) = \begin{cases} \text{Sand} & \text{hvis } v = 0 \\ \text{Falsk} & \text{hvis } v > 0 \ \wedge \ k = 0 \\ S(k-1, v) & \text{hvis } k > 0 \ \wedge \ 0 < v < x_k \\ S(k-1, v) \ \vee \ S(k-1, v - x_k) & \text{hvis } k > 0 \ \wedge \ x_k \le v \end{cases}$$

De følgende 4 opgaver består i at udfylde 4 blokke i følgende algoritmeskabelon.

Algoritme SubsetSum(L, V)
  $n = |L|$
  Opret tom tabel $T[0..n][0..V]$
  | **for** … **for** … ≪ **Opgave 21**: iterer over T ≫ |
    | ≪ **Opgave 22**: beregn $T[k][v] = S(k, v)$ ≫ |
  | ≪ **Opgave 23**: sæt *solution* til True eller False ≫ |
  if *solution* **then**
    | ≪ **Opgave 24**: Udskriv en løsning ≫ |
  else
    print "Not possible"

Opgave 21 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

Ja $\boxed{A}$	Ja $\boxed{A}$	Ja $\boxed{A}$
Nej $\boxed{B}$	Nej $\boxed{B}$	Nej $\boxed{B}$
for $k = 0$ **to** n	**for** $k = n$ **to** 0 **step** -1	**for** $k = 0$ **to** n
for $v = 0$ **to** V	**for** $v = 0$ **to** V	**for** $v = V$ **to** 0 **step** -1

Opgave 22 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

Ja [A]
Nej [B]

if $v = 0$ **then**
  $T[k][v] = \text{True}$
else if $v > 0$ **and** $k = 0$ **then**
  $T[k][v] = \text{False}$
else if $k > 0$ **and** $v < x_k$ **then**
  $T[k][v] = T[k-1][v]$
else
  $T[k][v] = T[k-1][v]$ **or** $T[k-1][v - x_k]$

Ja [A]
Nej [B]

$T[k][v] = T[k-1][v]$ **or** $T[k-1][v - x_k]$

Ja [A]
Nej [B]

if $v = 0$ **or** $(k > 0$ **and** $(T[k-1][v]$ **or** $(L[k] \leq v$ **and** $T[k-1][v - L[k]])))$ **then**
  $T[k][v] = \text{True}$
else
  $T[k][v] = \text{False}$

Opgave 23 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

Ja [A]
Nej [B]

solution = False
for $k = 0$ **to** n
  if $T[k][V]$ **then**
    solution = True

Ja [A]
Nej [B]

solution = False
for $v = 0$ **to** V
  if $T[n][v]$ **then**
    solution = True

Ja [A]
Nej [B]

solution = $T[n][V]$

Opgave 24 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

Ja [A]
Nej [B]

for $k = 1$ **to** n
  if $T[k][V]$ **then**
    print x_k

Ja [A]
Nej [B]

$v = V$
for $k = n$ **to** 1 **step** -1
  if $x_k \leq v$ **and** $T[k-1][v - x_k]$ **then**
    print x_k
    $v = v - x_k$

Ja [A]
Nej [B]

$v = V$
while $v > 0$
  for $k = 1$ **to** n
    if $T[k][v]$ **then**
      print x_k
      $v = v - x_k$