

EKSAMEN

Algoritmer og Datastrukturer

Lørdag den 4. januar 2020, kl. 9.00–11.00

Institut for Datalogi, Naturvidenskabelige Fakultet, Aarhus Universitet

Antal sider i opgavesættet (incl. forsiden): 16

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 (Asymptotisk notation, 6 %)

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

	Ja	Nej
$\log(n!)$ er $O(\log n^2)$	<input type="checkbox"/> A	<input type="checkbox"/> B
$n^{2/3} + n^3$ er $O(n^2)$	<input type="checkbox"/> A	<input type="checkbox"/> B
$5^5 + n \cdot \log n$ er $O(n^{2/3})$	<input type="checkbox"/> A	<input type="checkbox"/> B
$5 \cdot 2^{2 \log n}$ er $O((\log n)^3)$	<input type="checkbox"/> A	<input type="checkbox"/> B
$\log n$ er $O(n \cdot \log n)$	<input type="checkbox"/> A	<input type="checkbox"/> B
$(\log n)^3 + 2^{\log n}$ er $O(n^{0.01})$	<input type="checkbox"/> A	<input type="checkbox"/> B
$n!$ er $O(\sqrt{n})$	<input type="checkbox"/> A	<input type="checkbox"/> B
$n^{0.001}$ er $O(1)$	<input type="checkbox"/> A	<input type="checkbox"/> B
3^n er $O(2^{3 \log n})$	<input type="checkbox"/> A	<input type="checkbox"/> B
n^2 er $\Theta(2^{2 \log n})$	<input type="checkbox"/> A	<input type="checkbox"/> B
4^4 er $\Omega((\log n)^6)$	<input type="checkbox"/> A	<input type="checkbox"/> B
$n!/4$ er $\Omega(n^3)$	<input type="checkbox"/> A	<input type="checkbox"/> B

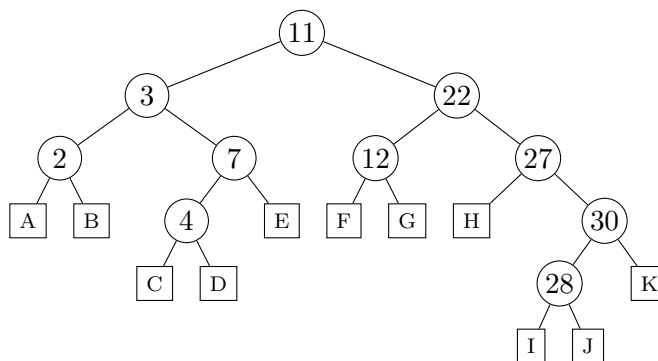
Opgave 2 (Analyse af løkker, 6 %)

<p>Algoritme loop1(n) $s = 1$ for $i = 1$ to n for $j = 1$ to n $s = s + 1$</p>	<p>Algoritme loop2(n) $s = 1$ while $s \leq n$ $s = s + 1$</p>
<p>Algoritme loop3(n) $s = 0$ for $i = 1$ to n for $j = i$ to n for $k = i$ to j $s = s + 1$</p>	<p>Algoritme loop4(n) $s = 0$ $i = n$ while $i > 1$ for $j = 1$ to i $s = s + 1$ $i = \lfloor i/2 \rfloor$</p>

Angiv for hver af ovenstående algoritmer udførelstiden som funktion af n i Θ -notation.

	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n\sqrt{n})$	$\Theta(n^3)$	$\Theta(n \log n)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta(\log \log n)$
loop1	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G	<input type="checkbox"/> H
loop2	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G	<input type="checkbox"/> H
loop3	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G	<input type="checkbox"/> H
loop4	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G	<input type="checkbox"/> H

Opgave 3 (Indsættelser i søgetræer, 4 %)



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 23, 13, 31, 14 og 6 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K
INSERT(23)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(13)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(31)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSERT(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Opgave 4 (Max-Heap-Insert, 4 %)

Angiv den binære max-heap efter indsættelse af elementerne 3, 5, 6, 4, 8, 7 og 11 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7	
11	8	7	4	5	3	6	<input type="checkbox"/> A
1	2	3	4	5	6	7	
3	5	6	4	8	7	11	<input type="checkbox"/> B
1	2	3	4	5	6	7	
11	6	8	3	4	5	7	<input type="checkbox"/> C
1	2	3	4	5	6	7	
11	8	7	6	5	4	3	<input type="checkbox"/> D
1	2	3	4	5	6	7	
6	8	11	4	5	7	3	<input type="checkbox"/> E

Opgave 5 (Build-Max-Heap, 4 %)

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

Hvad er resultat af BUILD-MAX-HEAP på ovenstående array ?

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

A

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

B

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

C

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

D

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

E

Opgave 6 (Heap-Extract-Max, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	24	14	13	21	22	9	2	6	10	20	18

Hvad er resultat af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12
25	14	24	9	13	21	22	18	2	6	10	20

A

1	2	3	4	5	6	7	8	9	10	11	12
25	18	24	14	13	21	22	9	2	6	10	20

B

1	2	3	4	5	6	7	8	9	10	11	12
25	14	24	9	13	21	22	2	6	10	20	18

C

1	2	3	4	5	6	7	8	9	10	11	12	13
25	14	24	9	13	21	22		2	6	10	20	18

D

1	2	3	4	5	6	7	8	9	10	11	12
25	24	22	13	21	18	9	2	6	10	20	14

E

Opgave 7 (Partition, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	3	19	29	23	16	6	12	25	9	13	14	20	24

Angiv resultatet af at anvende PARTITION($A, 4, 14$) på ovenstående array A .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	6	9	12	13	14	16	19	20	23	24	25	29

 A

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	3	19	16	6	12	9	13	14	20	29	25	23	24

 B

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	3	19	16	6	12	9	13	14	20	29	23	25	24

 C

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	5	3	6	9	12	13	14	16	19	20	23	25	29	24

 D

Opgave 8 (Radix-sort, 4 %)

3441 1412 3421 0012 3341 2141

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4, k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

1412 0012 3421 3441 3341 2141 A

0012 1412 2141 3341 3421 3441 B

0012 1412 3421 2141 3341 3441 C

0012 1412 2141 3341 3441 3421 D

3421 3441 3341 2141 1412 0012 E

Opgave 9 (Lineær probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
11							17	6	10	21

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \text{ mod } 11$.

Angiv positionerne de fem elementer 0, 3, 7, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 6, 10, 11, 17 og 21).

	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(3)	A	B	C	D	E	F	G	H	I	J	K
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K
INSERT(8)	A	B	C	D	E	F	G	H	I	J	K
INSERT(9)	A	B	C	D	E	F	G	H	I	J	K

Opgave 10 (Kvadratisk probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
	4	12			19			15	8	

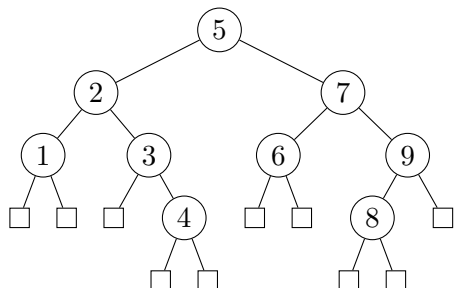
I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 2k \text{ mod } 11$ og $h = (h'(k) + 3i + 5i^2) \text{ mod } 11$.

Angiv positionerne de fem elementer 0, 5, 6, 7 og 10 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 4, 8, 12, 15 og 19).

	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(6)	A	B	C	D	E	F	G	H	I	J	K
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K
INSERT(10)	A	B	C	D	E	F	G	H	I	J	K

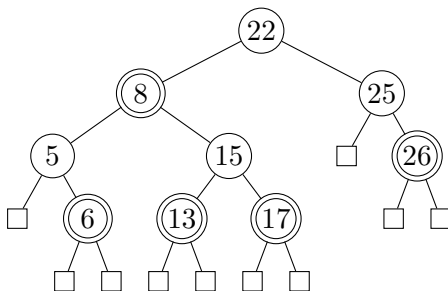
Opgave 11 (Rød-sort træ, 4 %)

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

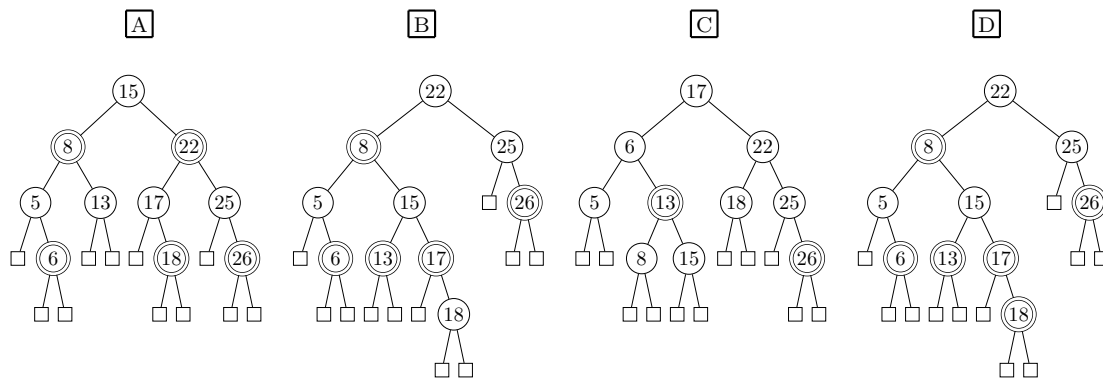


- | | Ja | Nej |
|---------------|----------------------------|----------------------------|
| 1, 3, 4, 7, 8 | <input type="checkbox"/> A | <input type="checkbox"/> B |
| 4, 5, 8 | <input type="checkbox"/> A | <input type="checkbox"/> B |
| 2, 4, 7, 8 | <input type="checkbox"/> A | <input type="checkbox"/> B |
| 4, 8 | <input type="checkbox"/> A | <input type="checkbox"/> B |
| 2, 4, 6, 8, 9 | <input type="checkbox"/> A | <input type="checkbox"/> B |

Opgave 12 (Indsættelse i rød-sort træer, 4 %)



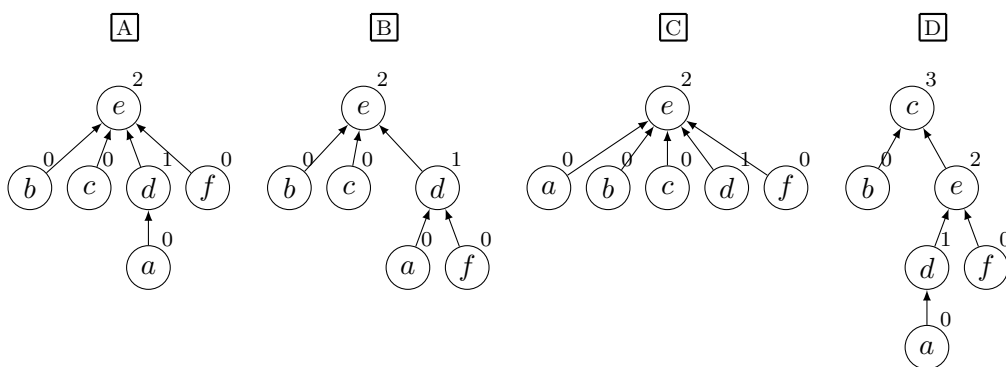
Angiv det resulterende rød-sortede træ når man indsætter 18 i ovenstående rød-sortede træ (dobbeltcirkler angiver røde knuder).



Opgave 13 (Union-find, 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(*f*, *d*)
- UNION(*a*, *d*)
- UNION(*b*, *e*)
- UNION(*a*, *e*)
- UNION(*f*, *c*)
- FIND-SET(*b*)

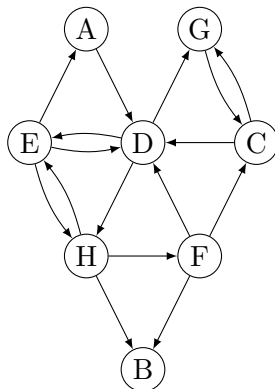


Opgave 14 (Rekursionsligninger, 4 %)

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$
$T(n) = 3 \cdot T(n/9) + 1$	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G
$T(n) = T(n - 1) + n$	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G
$T(n) = 2 \cdot T(n/4) + 1$	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G
$T(n) = 3 \cdot T(n/4) + n$	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G
$T(n) = 4 \cdot T(n/2) + n^2$	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E	<input type="checkbox"/> F	<input type="checkbox"/> G

Opgave 15 (BFS, 4 %)



For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver udtaget af køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A

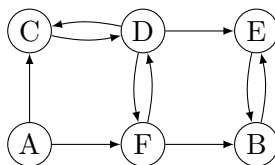
B

C

D

ADEGHCBF ADGHECFB ADEGHCFB ADEHBF CG

Opgave 16 (Lovlige bredde først træer, 4 %)



Angiv for hver af nedenstående mængder af kanter om de udgør et lovligt BFS træ for et bredde først gennemløb af ovenstående graf **startende i knuden A** og for en vilkårlig ordning af grafens incidenslister.

Ja Nej

(A,C) (A,F) (B,E) (F,B) (F,D) A B

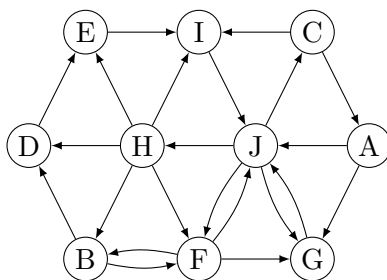
(A,C) (A,F) (C,D) (D,E) (F,B) A B

(A,C) (A,F) (D,E) (E,B) (F,D) A B

(A,C) (C,D) (D,E) (D,F) (F,B) A B

(A,F) (D,C) (D,E) (E,B) (F,D) A B

Opgave 17 (DFS, 4 %)



Betragt et dybde først gennemløb (DFS) 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 **finishing time**.

A

B

C

D

ICEDBFHJGA IEDBGFHCJA HEDBFICJGA EDBIHFCJGA

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

Tree edge Back edge Cross edge Forward edge

(B, D)

A

B

C

D

(J, H)

A

B

C

D

(I, J)

A

B

C

D

(H, B)

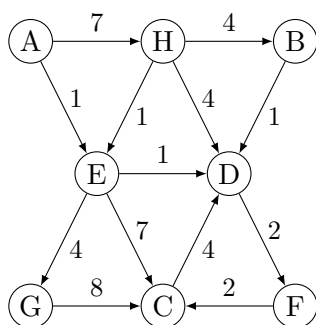
A

B

C

D

Opgave 18 (Dijkstras algoritme, 4 %)



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

A

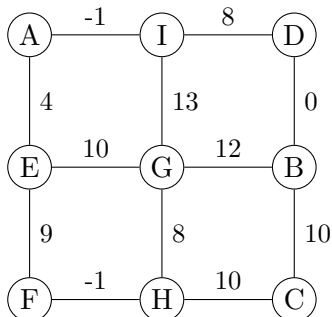
B

C

D

AECDFGHB AEHC DG BF AEDFGCHB AEDFCGHB

Opgave 19 (Prims algoritme, 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

B

C

D

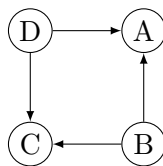
AIEDBGFHC

AIEDBFHCG

AIEDBFHGC

AIDBCHFEG

Opgave 20 (Topologisk sortering, 4 %)



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

Ja Nej

CBAD A B

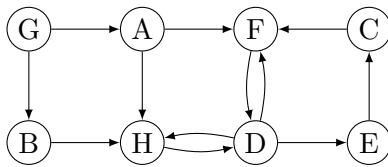
BDAC A B

DBAC A B

BCAD A B

DBCA A B

Opgave 21 (Stærke sammenhængskomponenter, 4 %)



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

- | | | | | | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> A | <input type="checkbox"/> B | <input type="checkbox"/> C | <input type="checkbox"/> D | <input type="checkbox"/> E | <input type="checkbox"/> F | <input type="checkbox"/> G | <input type="checkbox"/> H |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Sættet fortsætter på næste side!!!

Dynamisk programmering

De næste tre opgaver vedrører dynamisk programmering.

Professor Brodal har bedt Larsen om at lave et eksamenssæt til Algoritmer og Datastrukturer. Brodal har været så hjælpsom at lave et Python script der kan autogenerere opgaver. Når scriptet køres, genererer det m forskellige opgaver. Hver opgave giver et antal point, hvor point altid er et positivt heltal. Pointene for de forskellige opgaver gives som en liste $P = (p_1, \dots, p_m)$. Eksamenssættet skal indeholde præcist t point i alt. Larsen er i et ondt lune og vil gøre eksamenssættet så svært som muligt. Han har derfor vurderet de m opgaver og givet hver en sværhedsgrad $S = (s_1, \dots, s_m)$, hvor hver sværhedsgrad er et positivt tal. Larsen vil altså finde en delmængde af opgaver, således at det samlede antal point er præcist t og summen af sværhedsgrader er så høj som mulig.

Hvis vi eksempelvis har $m = 4$ opgaver med point $P = (2, 2, 4, 5)$ og sværhedsgrader $S = (5, 3, 6, 20)$, så er der to måder at opnå $t = 4$ point i alt: Vælg de første 2 opgaver og få en sum af sværhedsgrader på $5 + 3 = 8$, eller vælg opgave 3 og opnå en samlet sværhedsgrad på 6. Den bedste (sværeste) løsning er altså at vælge de to første opgaver.

Larsen indser at problemet kan løses med dynamisk programmering. Lad $V(k, v)$ betegne den maksimale sum af sværhedsgrader som kan opnåes ved at udvælge en delsekvens af de første k opgaver, således at det samlede antal point for de valgte opgaver er præcist v . $V(k, v)$ kan bestemmes ved følgende rekursionsformel

$$V(k, v) = \begin{cases} 0 & \text{hvis } k = 0 \text{ og } v = 0 \\ -\infty & \text{hvis } k = 0 \text{ og } v > 0 \\ V(k-1, v) & \text{hvis } p_k > v \text{ eller } V(k-1, v-p_k) = -\infty \\ \max\{V(k-1, v), V(k-1, v-p_k) + s_k\} & \text{ellers} \end{cases}$$

Den samlede sværhedsgrad af det optimale (sværeste) eksamenssæt er altså $V(m, t)$. Hvis $V(m, t) = -\infty$ betyder det, at det ikke er muligt, at lave et eksamenssæt med præcist t point.

De følgende 3 opgaver består i at udfylde 3 blokke i følgende algoritmeskabelon.

Algoritme SværtEksamenssæt(P, S, t)

$m = |P|$

Opret tom tabel $T[0..m][0..t]$

for ... for ... << Opgave 22: iterer over T >>
--

<< Opgave 23: beregn $T[k][v] = V(k, v)$ >>

if $V(m, t) > -\infty$ **then**

<< Opgave 24: Udskriv en løsning >>

else

print "Not possible"

Opgave 22 (4 %)

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

Ja A

Ja A

Ja A

Nej B

Nej B

Nej B

for $k = 0$ **to** m
for $v = 0$ **to** t

for $v = 0$ **to** t
for $k = 0$ **to** m

for $k = 0$ **to** m
for $v = t$ **to** 0 **step** -1

Opgave 23 (4 %)

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

Ja A

Nej B

```

if  $k = 0$  then
  if  $v = 0$  then
     $T[k][v] = 0$ 
  else
     $T[k][v] = -\infty$ 
else if  $p_k > v$  or  $T[k - 1][v - p_k] = -\infty$  then
   $T[k][v] = T[k - 1][v]$ 
else
   $T[k][v] = \max\{T[k - 1][v], T[k - 1][v - p_k] + s_k\}$ 

```

Ja A

Nej B

```

if  $k > 0$  and  $p_k \leq v$  and  $T[k - 1][v - p_k] > -\infty$  then
   $T[k][v] = \max\{T[k - 1][v], T[k - 1][v - p_k] + s_k\}$ 
else if  $k > 0$  then
   $T[k][v] = T[k - 1][v]$ 
else if  $k = 0$  and  $v > 0$  then
   $T[k][v] = -\infty$ 
else
   $T[k][v] = 0$ 

```

Ja A

Nej B

```

if  $k > 0$  then
   $T[k][v] = T[k - 1][v]$ 
if  $k > 0$  and  $p_k \leq v$  and  $T[k - 1][v - p_k] > -\infty$  then
   $T[k][v] = \max\{T[k][v], T[k - 1][v - p_k] + s_k\}$ 
else if  $k = 0$  and  $v > 0$  then
   $T[k][v] = -\infty$ 
else
   $T[k][v] = 0$ 

```

Opgave 24 (4 %)

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

Ja A **for** $k = m$ **to** 1 **step** -1
Nej B **if** $p_k \leq t$ **and** $T[k-1][t-p_k] + s_k = T[k][t]$ **then**
 print Exercise k

Ja A **for** $k = 1$ **to** m
Nej B **if** $T[k][t] > -\infty$ **then**
 print Exercise k

$v = t$
Ja A **for** $k = m$ **to** 1 **step** -1
Nej B **if** $p_k \leq v$ **and** $T[k-1][v-p_k] = T[k][v] - s_k$ **then**
 print Exercise k
 $v = v - p_k$