Danmarks Tekniske Universitet

Skriftlig prøve, den 22. maj 2014.

Kursusnavn: Algoritmer og datastrukturer 1

Kursusnummer: 02105

Hjælpemidler: Skriftlige hjælpemidler. Det er **ikke** tilladt at medbringe lommeregner.

Varighed: 4 timer.

Vægtning: Opgave 1 - 24 %, Opgave 2 - 20 %, Opgave 3 - 16 %, Opgave 4 - 15 %, Opgave 5 - 25 %. Vægtning er kun approksimativ. Karakteren gives ud fra en helhedsvurdering.

Alle opgaver besvares ved at udfylde de indrettede felter nedenfor. Som opgavebesvarelse afleveres blot denne og de efterfølgende sider i udfyldt stand. Hvis der opstår pladsmangel kan man eventuelt benytte ekstra papir som så vedlægges opgavebesvarelsen.

Asymptotiske grænser skal angives så tætte som muligt. Medmindre andet er angivet er basen på alle logaritmer 2 og $\log^k n$ betegner $(\log n)^k$.

1 Kompleksitet

1.1 (6 %) Angiv for hver af nedenstående udsagn om de er korrekte:

	Ja	Nej
$3n^4 + 2n^3 = O(n^3)$	☐	☐
$4n + \log^3 n = \Omega(n)$	☐	☐
$\frac{1}{3}n^5 = \Omega(n^4)$	☐	☐
$(\frac{1}{3}n^5 + n^2 + n)n = \Omega(n^6)$	☐	☐
$2^{n+2} = \Theta(2^n)$	☐	☐

1.2 (6 %) Arranger følgende funktioner i voksende rækkefølge efter asymptotisk vækst. Dvs. hvis funktionen $g(n)$ følger umiddelbart efter funktionen $f(n)$ i din liste skal der gælde at $f(n) = O(g(n))$.

$n^{1/3} \log n$

$2^{\log n}$

$17\sqrt{n}$

$3n^2$

$\frac{1}{4} \log^4 n$

$n^{0.75}$

Svar: ____

1.3 (6 %) Angiv køretiden for nedenstående algoritme. Skriv dit svar i O-notation som funktion af n.

Algoritme 1 Alg1(n)

```
i = 1
while i ≤ n do
    j = 1
    while j ≤ n − 7 do
        j = j + 1
    end while
    i = i + 1
end while
```

Svar: ____________________

1.4 (6 %) Angiv køretiden for nedenstående algoritme. Skriv dit svar i O-notation som funktion af n.

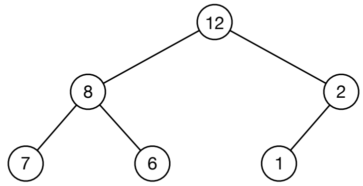
Algoritme 2 Alg2(n)

```
i = 1
j = 1
while i ≤ n do
    if j ≤ n then
        j = j + 1
    else
        j = 1
        i = i + 1
    end if
end while
```

Svar: ______________________

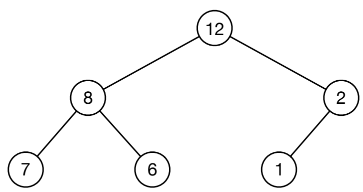
2 Datastrukturer

2.1 (4 %) Tegn hvordan nedenstående max-hob ser ud efter indsættelse af et element med nøgle 4.



Svar:

2.2 (4 %) Tegn hvordan nedenstående max-hob ser ud efter en EXTRACT-MAX operation.



Svar:

2.3 (4 %) Indsæt nøglesekvensen $70, 5, 2, 23, 80, 13, 26, 8$ i en hashtabel, der anvender hægtet hashing og hashfunktion $h(k) = k \mod 9$. Tegn indholdet af tabellen efter indsættelse af nøglesekvensen.

Svar:

2.4 (4 %) Indsæt nøglesekvensen $14, 48, 55, 74$ i en hashtabel, der anvender lineær probering og hashfunktion $k \mod 7$. Tegn indholdet af tabellen efter indsættelse af nøglesekvensen.

Svar:

2.5 (4 %) Vi vil gerne understøtte operationen MIN() på hver af de nedenstående fire datastrukturer. MIN() returnerer det mindste element i datastrukturen. F. eks. hvis datastrukturen indeholder elementerne $\{4, 6, 2, 19, 7\}$ returnerer MIN() værdien 2. Angiv for hver af datastrukturerne hvor lang tid det vil tage at udføre MIN() operationen i O-notation som funktion af n, hvor n er antallet af elementer i datastrukturen.

Tabel sorteret i stigende rækkefølge: ________________

Max-hob: ________________

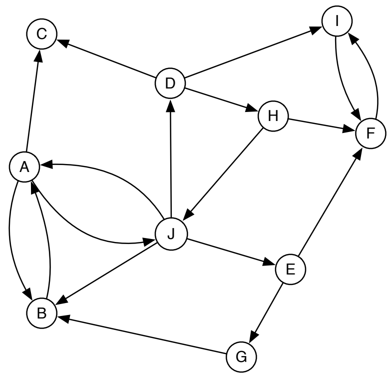
Min-hob: ________________

Usorteret tabel: ________________

Binært søgetræ: ________________

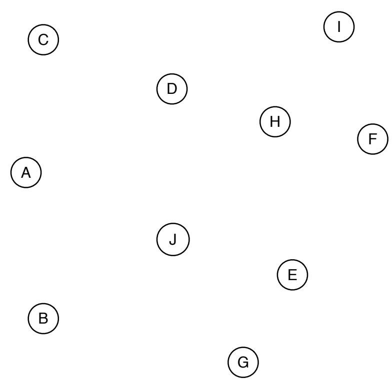
3 Grafer

I alle opgaverne antager vi at incidenslisterne er sorteret i alfabetisk orden. Betragt følgende graf G.



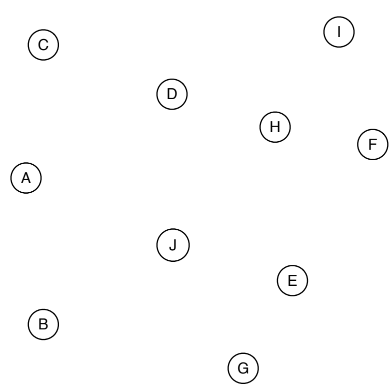
3.1 (4 %) Angiv et DFS træ for G med start i knude A.

Svar:

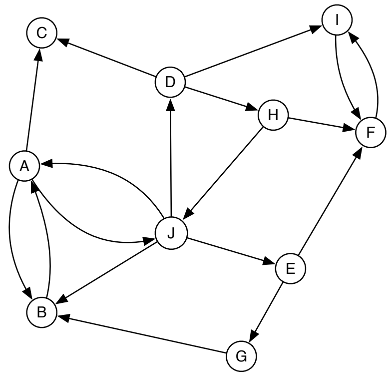


3.2 (4 %) Angiv et BFS træ for G med start i knude A. Angiv BFS-dybde/lag for hver knude. Antag incidenslisterne er sorteret i alfabetisk orden.

Svar:

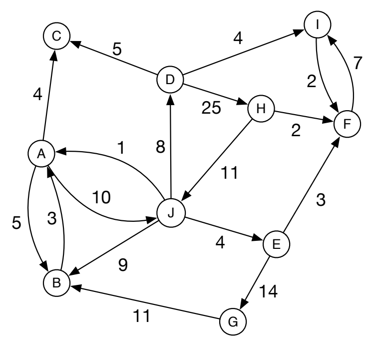


3.3 (4 %) Angiv de stærke sammenhængskomponenter i følgende graf. Det er tilstrækkeligt at opskrive opdeling af knuder.

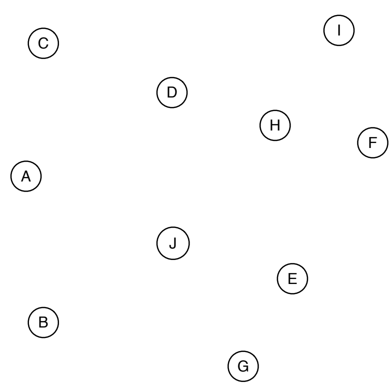


Svar:

3.4 (4 %) Angiv et korteste veje træ for følgende graf med start i knude *A*. Angiv længden af den korteste vej fra *A* til hver knude.



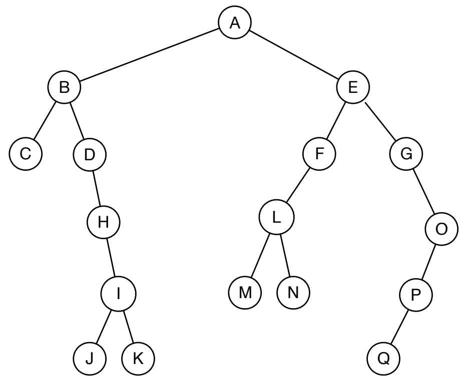
Svar:



4 Træer

Denne opgave omhandler rodfæstede binære træer. Enhver knude x har felterne $key[x]$, $left[x]$ og $right[x]$, der betegner hhv. nøglen, det venstre barn og det højre barn for x.

4.1 (1 %) En knude i et binært træ er en *kædeknude* hvis knuden har netop et barn og det barn ikke er et blad. Angiv alle kædeknuder i nedenstående træ.



Svar: ______________________________

4.2 (5 %) Giv en algoritme KÆDEKNUDE(x), der givet en knude x returnerer sand hvis og kun hvis x er en kædeknude. Din algoritme skal køre i konstant tid. Skriv din algoritme i pseudokode. Du må gerne antage at du har en konstant tids algoritme LEAF(x), der givet en knude x, returnerer sand hvis og kun hvis x er et blad.

Svar:

4.3 (9 %) Giv en rekursiv algoritme TÆLKÆDEKNUDER(X), der givet rodknuden returnerer antallet af kædeknuder i træet. Skriv din algoritme i pseudokode og analyser køretiden af din algoritme som funktion af n, hvor n er antallet af knuder i træet.

Svar:

5 Sociale netværk

Et *socialt netværk* er en mængde af P *personer* og V *venskabsrelationer*. Hver venskabsrelation består af to personer p_1 og p_2 og vi siger at p_1 og p_2 er *venner*. Hvis p_1 er ven med p_2 er p_2 ven med p_1. F. eks. er {Hans, Børge, Otto, Knud, Finn} og {(Hans, Børge), (Otto, Knud), (Børge, Otto), (Hans, Otto), (Finn, Knud)} er socialt netværk med 5 personer og 5 venskabsrelationer.

5.1 (2 %) Beskriv hvordan man kan modellere et socialt netværk som en graf.

Svar:

5.2 (1 %) Tegn grafen svarende til det sociale netværk i eksemplet.

Svar:

5.3 (5 %) En gruppe X af k personer kaldes et *tæt venskab* hvis alle personer i X er venner med alle andre personer i X. Giv en algoritme, der givet et socialt netværk og en mængde X af k personer, afgør om X er et tæt venskab eller ej. Analyser køretiden af din algoritme som funktion af P, V og k.

Svar:

5.4 (1 %) En *venskabskæde* af længde j er en sekvens af personer $p_1, \ldots, p_j$ så p_i er ven med p_{i+1} for alle i, $1 \leq i < j$. Givet en person p og et heltal $t \geq 0$ er p's t-*venner* alle personer, der har en venskabskæde til p af længde højst t (vi definerer 0-venner af p til at være p selv).

Angiv alle 2-venner af Hans i eksemplet. Svar: ____

5.5 (6 %) Giv en algoritme, der givet et socialt netværk, en person p og et heltal $t \geq 0$, beregner alle t-venner af p. Analyser køretiden af din algoritme som funktion af P, V og t.

Svar:

5.6 (6 %) Til enhver venskabsrelation r knytter vi en styrke *styrke*(r) som er et heltal større eller lig 0. Styrken af en venskabskæde er summen af styrkerne på hver venskabsrelation i venskabskæden. Giv en algoritme, der givet to personer p_1 og p_2 beregner den *svageste* venskabskæde mellem p_1 og p_2, dvs. beregner en venskabskæde af minimal styrke mellem p_1 og p_2. Analyser køretiden af din algoritme som funktion af P og V.

Svar:

5.7 (4 %) Antag nu at styrken på en venskabsrelation altid er et heltal mellem 1 og 10. Giv en algoritme, der løser problemet fra sidste spørgsmål mere effektivt i dette tilfælde. Analyser køretiden af din algoritme som funktion af P og V.

Svar: