Skriftlig Eksamen
DM507 Algoritmer og Datastrukturer

Institut for Matematik og Datalogi
Syddansk Universitet, Odense

Mandag den 24. juni 2013, kl. 10:00–14:00

Besvarelsen skal afleveres elektronisk. Se vejledning udsendt i kurset.

Alle hjælpemidler (lærebøger, notater, osv.) samt brug af computer er tilladt. Det er ikke tilladt at bruge internettet, undtagen til den elektroniske aflevering.

Eksamenssættet består af 7 opgaver på 7 nummererede sider (1–7). Fuld besvarelse er besvarelse af alle 7 opgaver. De enkelte opgavers vægt ved bedømmelsen er angivet i procent.

Der må gerne refereres til algoritmer og resultater fra lærebogen (Cormen et al., *Introduction to Algorithms*, 3rd edition), samt andre materialer fra kurset (f.eks. opgavesedler og slides). Henvisninger til andre kilder kan ikke bruges i besvarelsen af et spørgsmål.

Bemærk, at hvis der er et spørgsmål, man ikke kan besvare, må man gerne besvare de efterfølgende spørgsmål og blot antage, at man har en løsning til de foregående spørgsmål.

Opgave 1 (10%)

Angiv løsningen til hver af følgende rekursionsligninger.

$$\begin{array}{rl} \text{i)} & T(n) = 8 \cdot T(n/3) + n^2 \\ \text{ii)} & T(n) = 9 \cdot T(n/3) + n^2 \\ \text{iii)} & T(n) = 10 \cdot T(n/3) + n^2 \end{array}$$

Opgave 2 (10%)

Angiv for hvert af nedenstående udsagn, om de er sande eller falske.

$$\begin{array}{rl} \text{i)} & n \text{ er } O(n^{2/3}) \\ \text{ii)} & n \text{ er } O((3/2)^n) \\ \text{iii)} & 5n^7 + 7n^5 \text{ er } O(n^6) \\ \text{iv)} & (\log n)^2 \text{ er } O(2^{\log n}) \\ \text{v)} & n^1 \text{ er } O(1) \\ \text{vi)} & 1/n \text{ er } O(\log n) \end{array}$$

Opgave 3 (12%)

Spørgsmål a (4%):

Indsæt først 1 og derefter 2 i nedenstående min-heap A. Angiv udseendet af heapen efter den sidste af de to indsættelser.

A:

2	4	5	8	7	6	6	9		

Svar ved at skrive elementerne i rækkefølge fra venstre mod højre.

Spørgsmål b (4%):

Nedenstående er en hashtabel H som bruger hashfunktionen

$$h'(x) = (3x + 2) \bmod 11$$

samt linear probing (i lærebogen kaldes $h'(x)$ for “auxiliary hash function” i denne sammenhæng).

	0	1	2	3	4	5	6	7	8	9	10
H:	3			59		23	38	53		72	87

Indsæt først 60 og dernæst 45, og angiv udseendet af hashtabellen efter den sidste af de to indsættelser.

Svar ved at skrive indholdet af H i rækkefølge fra venstre mod højre, med tomme pladser angivet som x.

Spørgsmål c (4%):

Udfør PARTITION(A,1,8) på nedenstående array.

	1	2	3	4	5	6	7	8
A:	4	7	1	5	8	2	5	5

Svar ved at skrive indholdet af A efter udførslen i rækkefølge fra venstre mod højre.

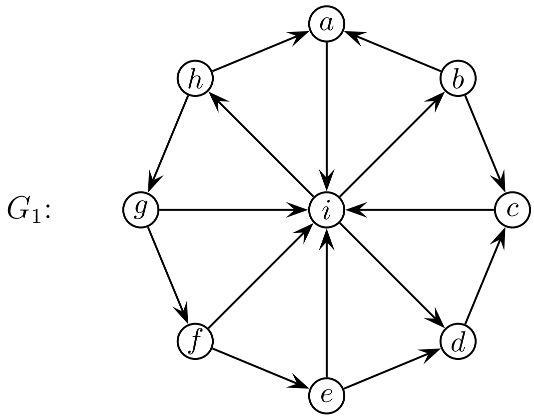
Opgave 4 (30%)

Spørgsmål a (10%):

Udfør BFS på grafen G_1 nedenfor, med start i knuden a.

Som svar, angiv knuderne i den rækkefølge de udtages fra køen (med operationen DEQUEUE) under algoritmens kørsel. Resultatet afhænger af ordningen af knuders nabolister. Du skal her antage at en knudes naboliste er sorteret i alfabetisk orden efter naboknudernes navne.

Angiv også for hver knude v slutværdien af $v.d$, dvs. afstanden fra a til v.

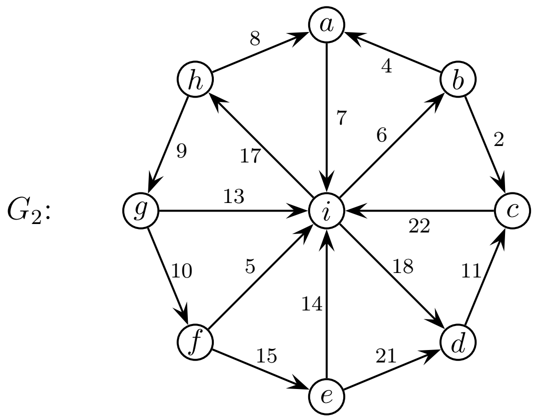


Spørgsmål b (10%):

Udfør Dijkstras algoritme på grafen G_2 nedenfor, med start i knuden a.

Som svar, angiv knuderne i den rækkefølge de udtages fra prioritetskøen (med operationen EXTRACT-MIN) under algoritmens kørsel.

Angiv også for hver knude v slutværdien af $v.d$, dvs. afstanden fra a til v.

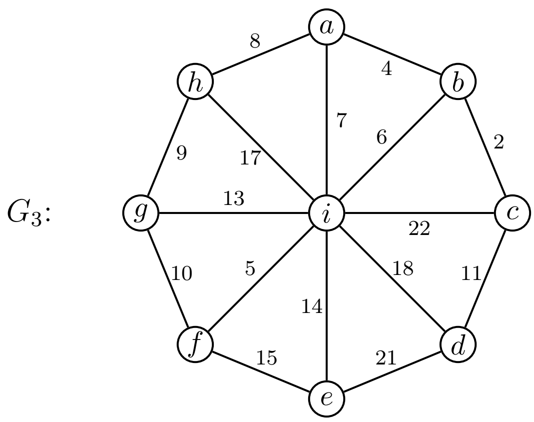


Spørgsmål c (10%):

Udfør Prims algoritme på grafen G_3 nedenfor, med start i knuden a.

Som svar, angiv knuderne i den rækkefølge de udtages fra prioritetskøen (med operationen EXTRACT-MIN) under algoritmens kørsel.

Angiv også kanterne i det resulterende minimum spanning tree (MST). En kant med endepunkter u og v skrives som sædvanligt (u, v).



Opgave 5 (10%)

Spørgsmål a (5%):

Vi ser her på at sortere n heltal med værdier mellem 0 og n^4. Angiv den asymptotiske worstcase køretid på denne type data for flg. sorteringsalgoritmer:

i) CountingSort

ii) RadixSort, når heltal betragtes som bestående af 4 digits med værdier mellem 0 og n.

iii) QuickSort

iv) MergeSort

v) InsertionSort

Spørgsmål b (5%):

Hvis CountingSort sorterer heltal med værdier mellem 0 og 7 (inkl.), hvad er da indholdet af array C *efter afslutningen* af algoritmen (med pseudo-kode som angivet i lærebogen side 195), når input A er som følger?

A:

7	4	1	2	6	4	0	4	4	4	7	2

Svar ved at angive elementerne i array C i rækkefølge fra venstre mod højre.

Opgave 6 (10%)

Følgende kode har til formål at beregne $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$ for heltal $n \geq 1$.

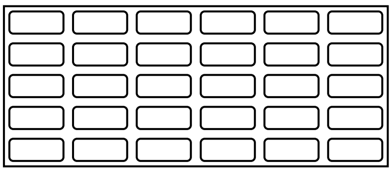
```
FACTORIAL(n)
    i = n
    r = 1
    while i > 1
        r = r * i
        i = i - 1
    return r
```

Angiv for hvert af nedenstående udsagn om det er en løkke-invariant for algoritmen (dvs. altid er sandt når testen i starten af **while**-løkken udføres) for input der er heltal $n \geq 1$. (Du behøver ikke argumentere for svarene.)

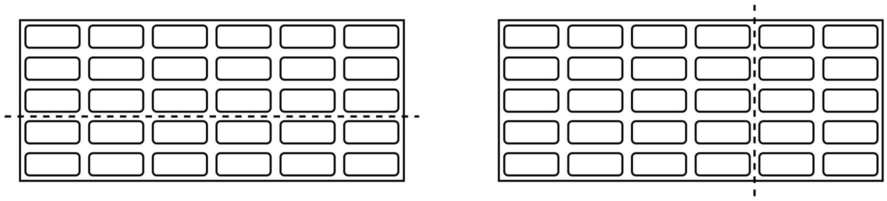
i) $i \geq 1$
ii) $r = i!$
iii) $r! \cdot i! = n!$
iv) $r = n!/i!$
v) $r = n!$

Opgave 7 (18%)

Chokolade sælges som oftest i firkantede plader med enkeltstykker støbt i i rækker og j søjler. Figuren herunder illustrerer en 5×6-plade.



Plader produceres på fabrikker i store $i \times j$-plader, mens forbrugerne værdsætter mindre plader. Du har et firma, som lever af at købe store plader og skære dem op i mindre. En plade kan skæres op med snit, som følger en rille mellem enten to rækker eller to søjler, og går tværs over hele pladen. Dette er illustreret herunder.



Efter et snit kan de opståede mindre plader igen skæres op med nye snit.

Du kender prisen, som en forbruger vil betale for de forskellige pladestørrelser, og ønsker at bestemme den bedst mulige opskæring af en stor plade i mindre plader.

I detaljer: Du ved at forbrugerne vil betale $P(k,l)$ for en $k \times l$-plade. Det giver for enhver opskæringen en samlet salgspris for stykkerne i opskæringen. Du ønsker for en $i \times j$-plade at bestemme den bedste samlede salgspris, der kan opnås ved en opskæring af pladen. Denne salgpris kaldes $C(i,j)$.

$C(i,j)$ kan beskrives ved følgende rekursive formel:

$$C(i,j) = \begin{cases} P(1,1) & \text{hvis } i = 1 \wedge j = 1 \\ \max\{P(i,1),\, \max_{1 \le s < i}\{C(s,1) + C(i-s,1)\}\} & \text{hvis } i > 1 \wedge j = 1 \\ \max\{P(1,j),\, \max_{1 \le t < j}\{C(1,t) + C(1,j-t)\}\} & \text{hvis } i = 1 \wedge j > 1 \\ \max\{P(i,j),\, \max_{1 \le s < i}\{C(s,j) + C(i-s,j)\}, & \\ \qquad\qquad \max_{1 \le t < j}\{C(i,t) + C(i,j-t)\}\} & \text{hvis } i > 1 \wedge j > 1 \end{cases}$$

Spørgsmål a (6%):

Givet værdierne for $P(k,l)$ vist i følgende tabel,

$k \setminus l$	1	2	3	4
1	1	2	3	4
2	2	5	7	9
3	3	8	10	12
4	5	8	17	21

beregn indholdet af følgende tabel for $C(i,j)$.

$i \setminus j$	1	2	3	4
1				
2				
3				
4				

Som svar, skriv 4×4 tal opstillet i tabellens orden.

Spørgsmål b (6%):

Beskriv en algoritme baseret på dynamisk programmering, der givet n, m, samt en tabel over $P(k,l)$ for $1 \leq k \leq n$, $1 \leq l \leq m$, beregner $C(n,m)$. Angiv algoritmens asymptotiske køretid og pladsforbrug som funktion af n og m.

Spørgsmål c (6%):

Argumenter for at den rekursive formel for $C(i,j)$ er korrekt.