Skriftlig Eksamen
DM507 Algoritmer og Datastrukturer

Institut for Matematik og Datalogi
Syddansk Universitet, Odense

Tirsdag den 24. juni 2014, kl. 10:00–14:00

Besvarelsen skal afleveres elektronisk. Se vejledning udsendt i kurset.

Alle skriftlige 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 10 opgaver på 7 nummererede sider (1–7). Fuld besvarelse er besvarelse af alle 10 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*, 3. udgave), 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 (8%)

Angiv løsningerne til følgende rekursionsligninger.

$$\begin{array}{rl} \text{i)} & T(n) = 2 \cdot T(n/3) + n \\ \text{ii)} & T(n) = 32 \cdot T(n/4) + n^{2.5} \end{array}$$

Opgave 2 (10%)

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

$$\begin{array}{rl} \text{i)} & n^2 \text{ er } O(n^2) \\ \text{ii)} & n^2 \text{ er } \Theta(n^2) \\ \text{iii)} & n^4 \text{ er } O(5n^3 + 3n^5) \\ \text{iv)} & n^4 \text{ er } \Theta(5n^3 + 3n^5) \\ \text{v)} & n \log n \text{ er } O(n^{1.5}) \\ \text{vi)} & n \text{ er } O(\log n) \\ \text{vii)} & (\log n)^{10} \text{ er } O(n^{0.10}) \\ \text{viii)} & 1 \text{ er } O(n) \\ \text{ix)} & n^2 \text{ er } o(n^3) \\ \text{x)} & n^3 \text{ er } \omega(n^3) \end{array}$$

Opgave 3 (6%)

Angiv udseendet af nedenstående array efter at have udført Build-Max-Heap på det.

1	2	3	4	5	6	7	8	9	10
5	4	3	2	1	10	9	8	7	6

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

Opgave 4 (6%)

Nedenstående er en hashtabel H der bruger double hashing med de to auxiliary hashfunktioner

$$h_1(x) = (5x + 1) \bmod 13$$
$$h_2(x) = 1 + (x \bmod 12)$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
H:	18		8			6			30	25		2	23

Indsæt værdierne 3, 5 og 15 (i den rækkefølge). Angiv udseendet af hashtabellen efter den sidste af de tre indsættelser.

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

Opgave 5 (6%)

Vi ønsker at bruge RADIX-SORT(A,4) til at sortere nedenstående array i stigende orden.

	1	2	3	4	5	6	7	8
A:	8345	7112	1830	5001	4345	2222	9112	6363

Vis indholdet af indholdet af A efter udførelsen af *tre* af de fire iterationer i RADIX-SORT(A,4).

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

Opgave 6 (6%)

En fil indeholder nedenstående tegn med de angivne hyppigheder. Der er 1900 tegn i alt.

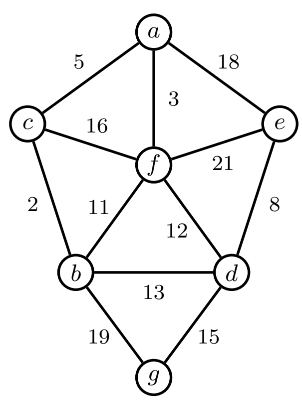
Tegn	`a`	`e`	`i`	`o`	`u`	`y`
Hyppighed	400	750	300	150	200	100

Lav et Huffman-træ på dette input. Angiv det resulterende kodeord for hvert af tegnene `a`, `e`, `i`, `o`, `u` og `y`, og angiv også hvor mange bits den kodede fil fylder (dvs. angiv den samlede længde af de 1900 kodede tegn).

Opgave 7 (17%)

Denne opgave handler om at bruge Kruskals algoritme til at finde et MST for nedenstående graf $G = (V, E)$. Vi ser i spørgsmål **a**, **b** og **d** på situationen efter at algoritmen har undersøgt 7 kanter (dvs. har lavet 7 iterationer af det andet **for**-loop på side 631 i lærebogen).

G:



Spørgsmål a (4%):

Angiv hvilke kanter der er valgt til at indgå i MST'et (dvs. er i A) efter at Kruskals algoritmen har undersøgt 7 kanter.

En kant med endepunkter u og v skrives som sædvanligt (u, v). I hver kant, angiv endepunkterne i alfabetisk rækkefølge.

Spørgsmål b (4%):

Angiv sammenhængskomponenterne som kanterne fra spørgsmål **a** giver anledning til, dvs. angiv sammenhængskomponenterne i grafen $G' = (V, A)$.

Hver sammenhængskomponent angives som en liste af knuder. I hver liste, angiv endepunkterne i alfabetisk rækkefølge.

Spørgsmål c (4%):

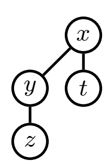
Angiv vægten af et minimum udspændende træ (MST) for hele grafen G.

Spørgsmål d (5%):

Vi antager nu at Kruskal bruger en disjoint-set datastruktur der er implementeret via en skov af træer som i lærebogens afsnit 21.3, under brug af både union-by-rank og path-compression heuristikken. Hvis der under UNION laves et LINK(x,y) på to knuder x og y med samme rank, antages det i dette spørgsmål at knuden med det alfabetisk mindste navn bliver den nye rod.

Angiv udseendet af disjoint-set skoven efter at Kruskals algoritme har undersøgt 7 kanter.

Hvert træ i skoven angives ved at skrive en liste af kanterne i det, samt hvilken knude som er roden. Angiv også rangen af roden. For eksempel kan følgende træ



angives således, hvis roden har rang 2:

$$(x,y),\ (y,z),\ (x,t),\ \text{rod} = x,\ \text{rang} = 2.$$

Opgave 8 (10%)

I denne opgave ser vi på at sortere n elementer efter værdien af deres nøgler, når det vides at disse nøgler kun antager værdierne 0 og 1.

Angiv for hver af algoritmerne CountingSort, InsertionSort, MergeSort og QuickSort, hvilke af nedenstående køretider som er henholdsvis deres worst-case og deres best-case køretid for denne type input.

A) $O(n)$
B) $O(n \log n)$
C) $O(n^2)$

Svar ved at angive indholdet (enten A, B eller C) af indgangene i følgende tabel:

	Worst case	Best case
CountingSort		
InsertionSort		
MergeSort		
QuickSort		

Opgave 9 (16%)

Angiv for hver af følgende algoritmer deres asymptotiske køretid i O-notation som funktion af n.

```
ALGORITME1(n)
    s = 0
    for i = 1 to n
        for j = i to n
            s = s + 1
```

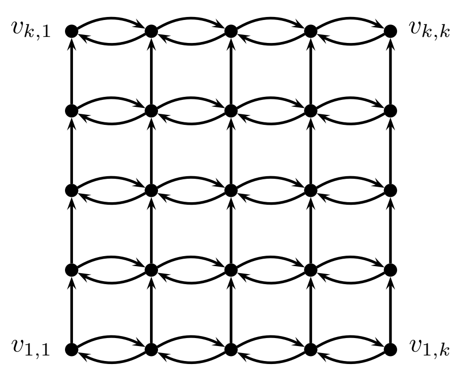
```
ALGORITME2(n)
    for i = 1 to n
        s = n
        while s > 1
            s = ⌊s/2⌋
```

```
ALGORITME3(n)
    s = 0
    for i = 1 to n
        for j = i to n
            for k = i to j
                s = s + 1
```

```
ALGORITME4(n)
    s = 0
    while n > 1
        for i = 1 to n
            s = s + 1
        n = ⌊n/2⌋
```

Opgave 10 (15%)

En *kvadrat-graf* er en orienteret graf med k rækker, hver med k knuder, og med kanter som illustreret i figuren nedenfor (for $k = 5$).



Mere præcist har en kvadrat-graf knuder $v_{i,j}$ for $i = 1, 2, \dots, k$ (rækkenummer) og $j = 1, 2, \dots, k$ (søjlenummer), samt kanter $(v_{i,j}, v_{i+1,j})$, $(v_{i,j}, v_{i,j+1})$ og $(v_{i,j}, v_{i,j-1})$ for alle værdier af i, j for hvilke begge kantens knuder eksisterer.

I resten af denne opgave antager vi at alle kanterne i en kvadrat-graf har en ikke-negativ vægt.

Spørgsmål a (3%):

Lad n og m betegne henholdsvis antal knuder og antal kanter i en kvadrat-graf. Udtryk n og m som funktion af k.

Spørgsmål b (4%):

Angiv udførselstiden for Dijkstra's algoritme som funktion af k når den udføres på en kvadrat-graf med start i knuden $v_{1,1}$.

Spørgsmål c (8%):

Konstruér en algoritme som i tid $O(m)$ finder længden af de korteste veje fra knuden $v_{1,1}$ til alle øvrige knuder i en kvadrat-graf. Beskriv (i ord eller pseudokode) algoritmen, og argumenter for algoritmens køretid og korrekthed.

Hint: Lemma 24.15 (side 673) fra lærebogen kan være inspirerende.