Institut for Matematik og Datalogi
Syddansk Universitet, Odense

13. maj 2025
RF

Opgaver Uge 21

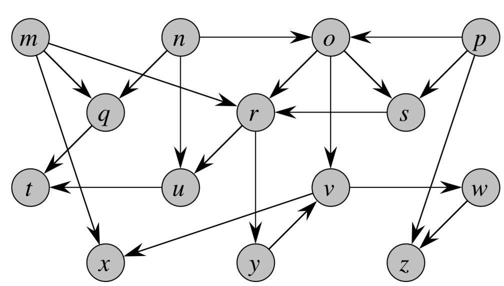
DM507/DM578/DS814/SE4-DMAD

Bemærk: Mange opgaver i denne uge er repetition af tidligere stof. F.eks. løses det meste af eksamenssættet fra juni 2014.

I.A: Løses i løbet af de første øvelsestimer i uge 21

1. Cormen et al., 4. udgave, øvelse 20.4-1 (side 575) [Cormen et al., 3. udgave: øvelse 22.4-1 (side 614)]:

   Kør TOPOLOGICAL-SORT(G) på nedenstående graf G og angiv den resulterende rækkefølge for knuderne. Antag, at **for**-loopet i linje 5 af DFS(G) løber gennem knuderne i alfabetisk rækkefølge, og at knuder i nabolister er sorteret i alfabetisk rækkefølge.



2. Opgave 1 fra eksamenssættet fra juni 2014:

Angiv løsningerne til følgende rekursionsligninger.

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

3. Opgave 2 fra eksamenssættet fra juni 2014:

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

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

4. Opgave 3 fra eksamenssættet fra juni 2014:

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.

5. Opgave 4 fra eksamenssættet fra juni 2014:

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`.

I.B: Løses hjemme inden de næste øvelsestimer i uge 21

1. Cormen et al., 4. udgave, øvelse 20.4-5 (side 576) [Cormen et al., 3. udgave: øvelse 22.4-5 (side 615)]:

   Her er en idé til en anden algoritme til at lave topologisk sortering af en DAG: sålænge der findes en knude v med indgrad nul, udskriv v og fjern v og den dens udgående kanter fra grafen. Argumentér for korrekthed af denne idé, dvs. for at udskrivningsrækkefølgen for knuderne udgør en topologisk sortering. Beskriv også en metode til at implementere idéen, så den resulterende algoritme kører i tid $O(|V| + |E|)$, og argumentér for denne køretid. Ekstraspørgsmål: hvad sker der i din algoritme, hvis inputgrafen har en kreds?

2. Opgave 5 fra eksamenssættet fra juni 2014:

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.

3. Opgave 6 fra eksamenssættet fra juni 2014:

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).

4. Opgave 8 fra eksamenssættet fra juni 2014:

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		

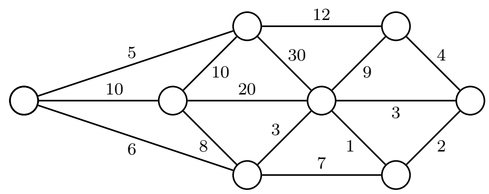
II.A: Løses i løbet af de næste øvelsestimer i uge 21

1. Cormen et al., 4. udgave, øvelse 20.5-1 (side 580) [Cormen et al., 3. udgave: øvelse 22.5-1 (side 620)]:

   Hvordan kan antallet af stærke sammenshængskomponenter ændre sig, hvis man til en orienteret graf tilføjer én ny kant? Besvar samme spørgsmål for sammenhængskomponenter i en uorienteret graf.

2. Eksamen juni 2010, opgave 2, spørgsmål d:

Spørgsmål d (6%): For grafen G_4, angiv et minimum spanning tree (MST), samt dets vægt.



Figur 4: Grafen G_4

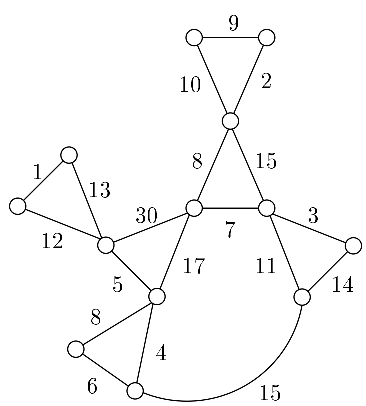
Du skal (til eksaminatorierne, ikke til eksamen) løse den både med Prim-Jarníks og Kruskals algoritme, og samt vise skridtene undervejs.

3. Eksamen januar 2008, opgave 2, spørgsmål b:

Spørgsmål b (8%): Tegn et letteste udspændende træ for nedenstående vægtede graf.

Du kan evt. bruge tegningen på sidste side.

Argumenter for, at dit resultat er et letteste udspændende træ.



Du skal (til eksaminatorierne, ikke til eksamen) løse den både med Prim-Jarníks og Kruskals algoritme, og samt vise skridtene undervejs. Du behøver ikke argumentere for korrekhed (dvs. se bort fra sidste linje i opgaven).

4. Cormen et al., 4. udgave, øvelse 21.2-4 (side 598) [Cormen et al., 3. udgave: øvelse 23.2-4 (side 637)]. Det er nok at svare på første spørgsmål:

   Hvor hurtig kan du få Kruskals algoritme til at køre, hvis alle kantvægte er heltal mellem 1 og $|V|$?

5. Opgave 9 fra eksamenssættet fra juni 2014:

   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⌋
```

II.B: Løses hjemme inden øvelsestimerne i uge 22

1. Eksamen juni 2012, opgave 2:

Spørgsmål a (5%):

Angiv hvilke af de fire arrays A_1, A_2, A_3 og A_4 som repræsenterer en min-heap.

A_1:	7	4	9	2	6	8	10	1	3	5
A_2:	1	2	3	4	5	6	7	8	9	10
A_3:	1	2	3	4	1	2	3	4	5	6
A_4:	1	1	1	1	1	1	1	1	1	1

$\square$

Spørgsmål b (5%):

Angiv udseendet af min-heapen A_5 efter udførelse af en HEAP-EXTRACT-MIN operation. (Svar ved at skrive elementer i rækkefølge fra venstre mod højre.)

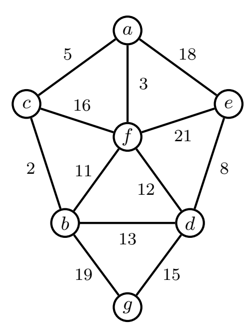
A_5:	1	2	5	3	7	9	6	8	4	10

$\square$

2. Eksamen juni 2014, opgave 7:

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%):

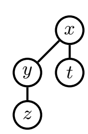
Angiv vægten af et minimum udspændende træ (MST) for hele grafen G.

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:

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

For 4. udgave af lærebogen: i teksten i starten skal der henvises til side 594 (i stedet for side 631) og i spørgsmål d skal henvisningen være til afsnit 19.3 (i stedet for afsnit 21.3).