Institut for Matematik og Datalogi
Syddansk Universitet, Odense

22. May 2025

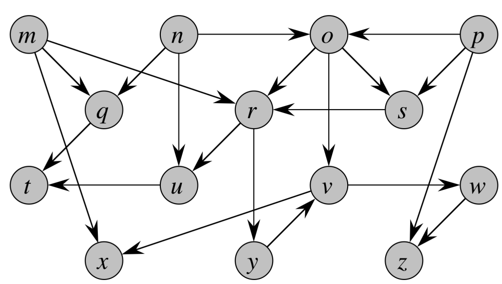
Exercise Sheet Week 14

DSK814

A(I). Solve in the practice sessions

1. Cormen et al., 4. Edition: Exercise 20.4-1 (page 575) [Cormen et al., 3. Edition: Exercise 22.4-1 (page 614)].

   Run TOPOLOGICAL-SORT(G) on the graph G below and specify the resulting order of the nodes. Assume that both the nodes considered in the **for** loop in line 5 of DFS(G) and the neighbor lists are sorted in alphabetical order.



2. Provide the solution to the following recurrences.

$$T(n) = 2 \cdot T(n/3) + n$$
$$T(n) = 32 \cdot T(n/4) + n^{2.5}$$

3. For each of the statements below, indicate whether they are true or false.

(a) $n^2 = O(n)$

(b) $n^2 = \Theta(n^2)$

(c) $n^4 = O(5n^3 + 3n^5)$

(d) $n^4 = \Theta(5n^3 + 3n^5)$

(e) $n \log n = O(n^{1.5})$

(f) $n = O(\log n)$

(g) $(\log n)^{10} = O(n^{0.1})$

(h) $1 = O(n)$

(i) $n^2 = o(n^3)$

(j) $n^3 = \omega(n^3)$

4. Provide the state of the array $[5, 4, 3, 2, 1, 10, 9, 8, 7, 6]$ after performing Build-MaxHeap.

5. Insert the values 3, 5 and 15 into the hash table H using double hashing and the two auxiliary hash functions

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

Provide the state of H after the last insertion.

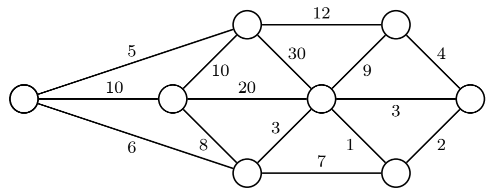
	0	1	2	3	4	5	6	7	8	9	10	11	12
H:	18		8			6			30	25		2	23

A(II). Solve in the practice sessions

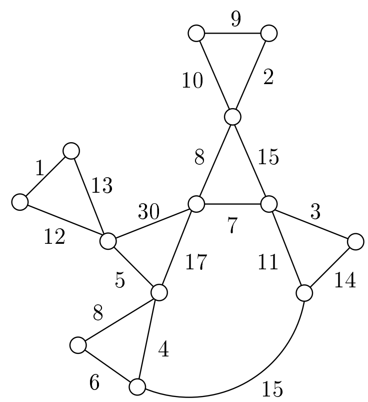
1. Cormen et al., 4. Edition: Exercise 20.5-1 (page 580) [Cormen et al., 3. Edition: Exercise 22.5-1 (page 620)].

   How much can the number of strong connected components change when adding a new edge to a directed graph?

2. Provide a MST for the graph below using both Prim-Jarník's and Kruskal's algorithm.



3. Provide a MST for the graph below using both Prim-Jarník's and Kruskal's algorithm.



4. Cormen et al., 4. Edition: Exercise 21.2-4 (page 598) [Cormen et al., 3. Edition: Exercise 23.2-4 (page 637)].

   What is the running time of Kruskal's algorithm if all edge weights are integers between 1 and $|V|$?

5. For each of the following algorithms, specify the asymptotic running time in O-notation as a function of n.

```
ALGORITHM1(n)                 ALGORITHM2(n)
    s = 0                         for i = 1 to n
    for i = 1 to n                    s = n
        for j = i to n                while s > 1
            s = s + 1                     s = ⌊s/2⌋
```

Algorithm3(n)
 $s = 0$
 for $i = 1$ **to** n
  for $j = i$ **to** n
   for $k = i$ **to** j
   $s = s + 1$

Algorithm4(n)
 $s = 0$
 while $n > 1$
  for $i = 1$ **to** n
   $s = s + 1$
  $n = \lfloor n/2 \rfloor$

B(I). Solve at home before tutorial in week 15

1. Cormen et al., 4. Edition: Exercise 20.4-5 (page 576) [Cormen et al., 3. Edition: Exercise 22.4-5 (page 615)].

   Another way to topologically sort a directed acyclic graph $G = (V, E)$ is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Explain how to implement this idea so that it runs in time $O(|V| + |E|)$. What happens to this algorithm if G has cycles?

2. Sort the array $[8345, 7112, 1830, 5001, 4345, 2222, 9112, 6363]$ in ascending order using Radix-Sort$(A, 4)$. Show the state of A after performing three of the five iterations of Radix-Sort$(A, 4)$.

3. A file with 1900 characters in total contains the following characters with the specified frequencies.

Character	a	e	i	o	u	y
Frequency	400	750	300	150	200	100

   Create a Huffman tree on this input and specify the resulting code words for the characters a, e, i, o, u and y. Argue how many bits the encoded file requires, i.e. specify the total length of the 1900 encoded characters.

4. Consider sorting n elements by their keys which are either 0 and 1. For this type of input, specify the worst and best case running times for the algorithms CountingSort, InsertionSort, MergeSort and QuickSort. Fill in the following table with $O(n), O(n \log n), O(n^2)$.

	Worst Case	Best Case
CountingSort		
InsertionSort		
MergeSort		
QuickSort		

B(II). Solve at home before tutorial in week 15

1. Specify which of the following four arrays A_1, A_2, A_3 and A_4 represent a min-heap.

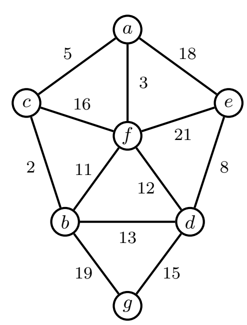
$$\begin{aligned} A_1: &\quad [7, 4, 9, 2, 6, 8, 10, 1, 3, 5] \\ A_2: &\quad [1, 2, 3, 4, 6, 7, 8, 9, 10] \\ A_3: &\quad [1, 2, 3, 4, 1, 2, 3, 4, 5, 6] \\ A_4: &\quad [1, 1, 1, 1, 1, 1, 1, 1, 1, 1] \end{aligned}$$

2. Provide the min-heap after performing Heap-Extract-Min on A.

$$A: \quad [1, 2, 3, 4, 6, 7, 8, 9, 10]$$

3. Consider using Kruskal's algorithm to find an MST for grah G below.

G:



   (a) State which edges are included in the MST after Kruskal's algorithm has examined 7 edges. (examined NOT selected)

   (b) Specify the connected components given by the edges in part (a).

   (c) Provide the weight of the MST for G.

(d) Assume that Kruskal algorithm uses a disjoint-set data structure that is implemented via a forest of trees using both the union-by-rank and path-compression heuristics. Whenever UNION uses LINK(x, y) on two nodes x and y of the same rank, the lexicographically smallest node becomes the new root.

State the disjoint-set forest after Kruskal's algorithm has examined 7 edges. Each tree in the forest is specified by listing its edges and the root and its rank. For example, the following tree is specified by $(x, y), (y, z), (x, t), \text{root} = x, \text{rang} = 2$.