

Algoritmer och datastrukturer DV2,

Exercise Exam

1.

Arrange the following functions in increasing order according to asymptotic growth.

$$3^N, 4^N, \sqrt{4^N}, \log^2 N, \sqrt{\log N}, \sqrt{N}, N^2, \log N, 20N, \frac{N}{\log N} \quad 2p$$

2.

Prove that if

$$x = \Theta(y / \log y)$$

then

$$y = \Theta(x \log x) \quad 4p$$

3.

Prove that if

$$T(n) = T(n^{1/3}) + 1, \quad T(2) = O(1)$$

then

$$T(n) = O(\log \log n) \quad 4p$$

4.

In a text, the following characters occur with specified frequencies. Show

- how the Huffman codes are computed, and the resulting code for each character 3p
- how the string ALLA is encoded. 2p

A	L	U	V	X	Y
3	18	2	1	4	17

5.

Dijkstra's algorithm for single source shortest paths maintains two sets of vertices

- the set S of vertices where the distance from the source is known
- a priority queue Q of the other vertices

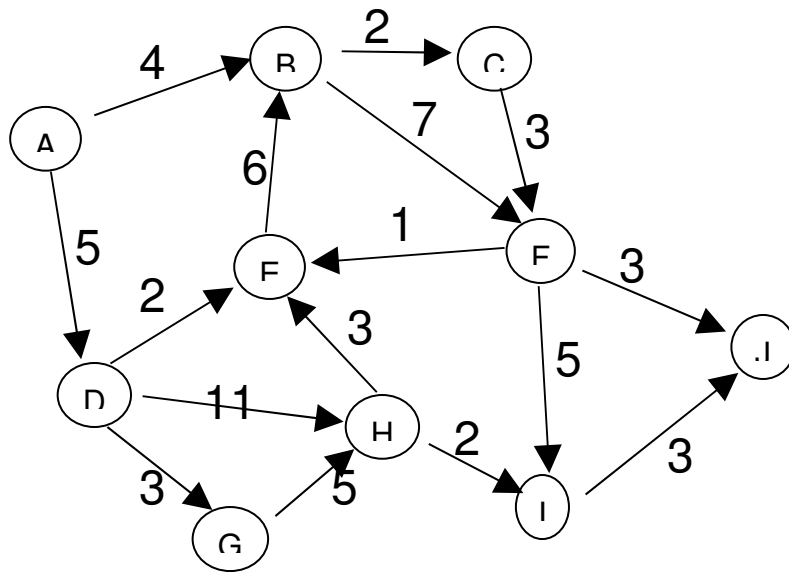
The priority queue Q has the operations *Initialize*, *Delete-Min*, and *Decrease-Key*

a) Write the vertices in the order as they are inserted into S when running Dijkstra's algorithm on the graph below with vertex A being the source. 4p

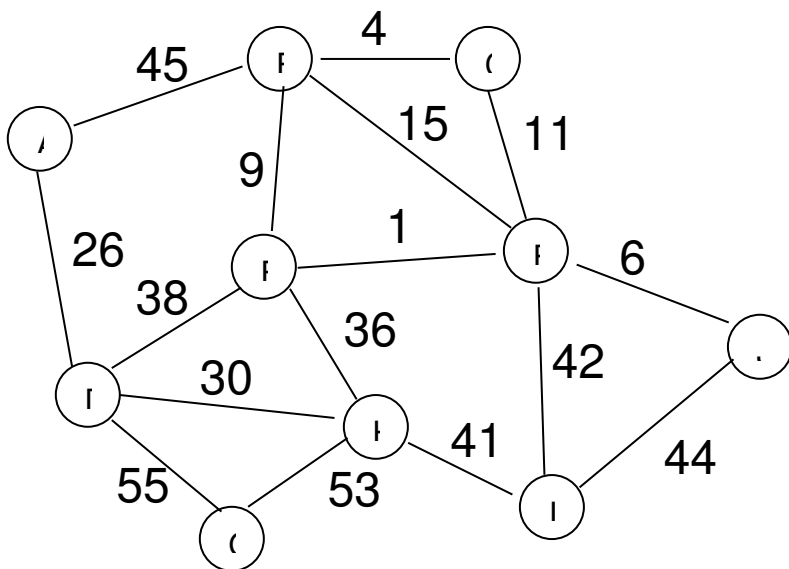
b) Describe an implementation of Dijkstra's algorithm that runs in $O((V + E) \log V)$ time for a graph with V vertices and E edges, and explain how the complexity is achieved. 4p

c) A *monotone* priority queue implements operations *Initialize*, *Delete-Min*, and *Decrease-Key* in the restricted case where the *Decrease-Key* operation is not allowed

to decrease keys to values that are smaller than the current min value. In fact, Dijkstra's algorithm only requires a monotone priority queue. Explain why. 4p



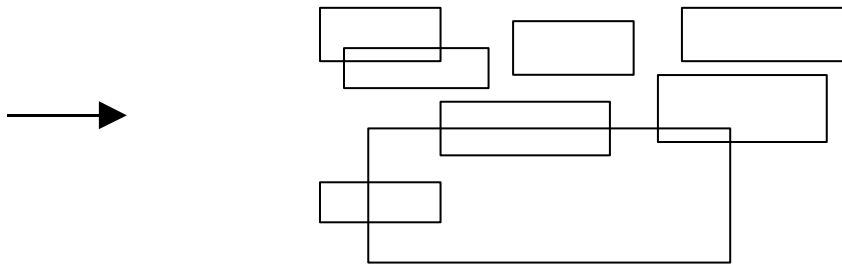
6. Write the edges (by writing the weight) in the order they are added when Kruskal's algorithm is run to compute a minimum spanning tree for the graph below. 4p



7. Consider a set of n rectangles as in the figure below, where each rectangle is represented by four numbers (the coordinates of the lower left and upper right corners). Further, consider an arrow arriving horizontally from left at position y . The arrow will pass through a number of rectangles (possibly zero) and then disappear to the right.

a) Describe an efficient algorithm for computing which rectangles the arrow will pass, and explain its complexity. 4p

b) Describe an efficient algorithm for computing which rectangle the arrow will hit first, if any, and explain its complexity. 4p



8.

Assume that we have n objects where each object has a size between 0 and 1. Consider the following problem (bin packing):

Distribute the n objects into as few bins as possible such that the total size in each bin is at most 1.

For example: if the elements are 0.1 0.5 0.8 0.2 0.85 0.9 0.25 the smallest number of bins is four, such as in the following solution:

(0.1 0.9) (0.5 0.2 0.25) (0.8) (0.85)

The problem is very hard to solve in general, but there are some heuristic algorithms that produce good results in practice. One of them is Best-Fit:

- Start with one empty bin:
- For each element, put it in the bin which leaves the least room left over. If the element can not be placed into any bin, add a new bin.

For the elements above, Best-Fit would run as follows

1. element 0.1 goes into the first bin
2. element 0.5 goes into the first bin
3. element 0.8 can not fit into the first bin, so we create a 2nd bin and place it there.
We now have two bins: (0.1 0.5) and (0.8)
4. element 0.2 have two bins to choose from, the best fit is to place it in the 2nd bin which gives us the bins as (0.1 0.5) and (0.8 0.2)
5. etc

Describe how to maintain the bins such that each step (adding an element to the best-fit bin) can be executed in $O(\log b)$ time, where b is the number of bins

4p