Algorithms - CS5800 - Summer 2, 2017 - Assignment 2

1. Describe a comparison algorithm Mid(a,b,n) with complexity $O(\log_2 n)$, that finds the median of the union of two sorted arrays.
   The input for the algorithm are two sorted arrays of length n:
   $$a = (a_1, a_2, ..., a_n) \text{ such that } a_1 < a_2 < ... < a_n$$
   $$b = (b_1, b_2, ..., b_n) \text{ such that } b_1 < b_2 < ... < b_n$$
   The output of the algorithm is the median of the union of the sets of elements of the arrays - $a \cup b$.
   Write the pseudocode of your algorithm and prove that the complexity is $O(\log_2 n)$.
   You may assume that no two elements are equal.
   Remark: In this question we define the median of m elements $a_1 < a_2 < ... < a_m$ to be $a_{\lceil \frac{m}{2} \rceil}$.
   It means that:
   The median of $1, 2, 3, 4, 5, 6, 7$ is 4
   The median of $1, 2, 3, 4, 5, 6, 7, 8$ is (again) 4.

2. Apply the algorithm we saw in the class, to find the longest common subsequence of the sequences:
   $$a = 4, 2, 6, 9, 0$$
   $$b = 2, 5, 9, 0, 4$$
   Show how the entries of the matrix are calculated.

3. Alice and Bob play the following game:
   Given a sequence of numbers:
   $$a_1, a_2, ..., a_n$$
   each player at her/his turn, takes either the first or last number of the sequence. (So if Alice takes a_1 at the first turn, then Bob can take either a_2 or a_n at his turn. If Bob takes a_n then Alice can take either a_2 or a_{n-1} and so on)
   The game ends when there are no numbers left.
   At the end of the game the score of each player is the sum of the numbers she/he took, minus the sum of the numbers of the opponent.
   Describe a dynamic programming algorithm that computes the maximal score that the first player can guarantee.
   What is the complexity of the algorithm that you suggest?

4. (a) Apply the Huffman algorithm to find an optimal prefix code for the probability vector:
   $$p = (0.05, 0.05, 0.1, 0.2, 0.25, 0.35)$$

Compute the average length of a codeword in the prefix code that you found.
(b) Find a probability vector $p = (p_1, ..., p_6)$ for the characters $y_1, ..., y_6$ such that the following prefix code is optimal:

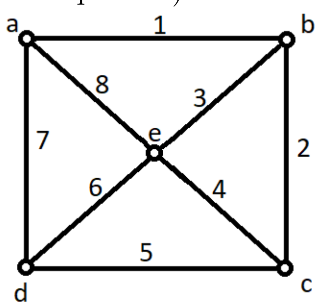
$$y_1 \to 01 \ , \ y_2 \to 001 \ , \ y_3 \to 000 \ , \ y_4 \to 11 \ , \ y_5 \to 100 \ , \ y_6 \to 101$$

(c) For a probability vector $p = (p_1, ..., p_n)$ we define:

$$f(p) = \sum_{i=1}^{n} (p_i \cdot l_i)$$

where $l_1, ..., l_n$ are the lengths of the codewords in an optimal prefix code for p.
Prove or disprove:
For any probability vector of the form $p = (p_1, p_2, ..., p_7)$ we have $f(p) < \frac{41}{14}$.

5. Apply Kruskal's algorithm for the following graph (the weights of the edges are in the picture):



Show the steps of the algorithm.

6. In this question you will show that if $G = (V, E)$ is a weighted graph such that no 2 edges have the same weight, then the minimal spanning tree is unique.

(a) Let T_1 and T_2 be two spanning trees of a graph $G = (V, E)$ (note that T_1 and T_2 are not MST. The edges have no weights at this point).
Let $e_1 \in E$ be an edge such that $e_1 \in T_1$ and $e_1 \notin T_2$.
Show that there exists an edge $e_2 \in T_2$ such that:
$(T_1 - \{e_1\}) \cup \{e_2\}$ and $(T_2 - \{e_2\}) \cup \{e_1\}$ are both spanning trees of G.

(b) Assume that the edges of the graph G have weights such that for any two different edges $e_i \neq e_j$ of G, we have $w(e_i) \neq w(e_j)$.
Show that there exists a **unique** minimal spanning tree for G.
Hint: Assume there are 2 different minimal spanning trees T_1 and T_2. Order their edges by their weights:

$$T_1 : w(e_1) < w(e_2) < ... < w(e_k)$$

$$T_2 : w(e'_1) < w(e'_2) < ... < w(e'_k)$$

Consider the maximal index i such that $w(e_i) \neq w(e'_i)$ and use the result from part (a).