Algorithm Design and Complexity – Exam

TIME: 80 minutes
POINTS: 40 points

A. Theory (20p)

(4p) A1. Starting from their definitions, prove that the set $A = O(f) \cap \Omega(f)$ is not empty. How can you best describe the functions in A ?

(4p) A2. For solving the activity selection problem, we can use either dynamic programming or greedy. Point out the recursive solution that describes the problem and how to transform the dynamic programming approach into a greedy one.

(4p) A3. Are there any similarities between BFS and Dijkstra's algorithm? What are the differences?

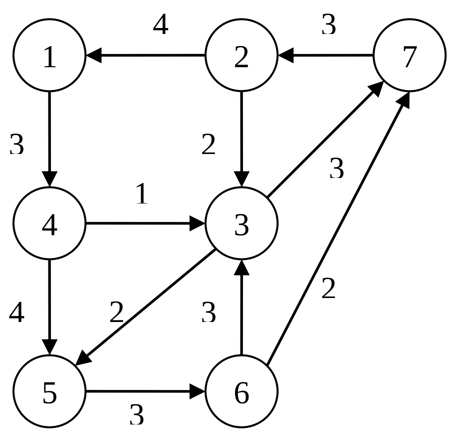
(4p) A4. A depth-first search of a directed graph can be used to classify its edges. Explain how and what types of edges do we have. Give a simple example.

(4p) A5. Which is the generic algorithm for computing the minimum spanning tree (MST) of a graph? Which are the differences between Kruskal's and Prim's algorithm for computing the MST.

B. Exercices (12p)

(6p) B1. Solve the following recurrences (choose two from below):

a. $T(n) = 4 * T(n/2) + n * n$
b. $T(n) = T(n/2) + T(n/4) + n^2$
c. $T(n) = T(n/2 + \text{sqrt}(n)) + n$ *(Hint! Try to guess a solution by solving a similar but simpler recurrence and prove it using the substitution method)*



(6p) B2. Considering the graph on the left, show how the following algorithms operate (choose two from below):

a. Ignoring the weights, run a BFS from node 1;
b. Compute the matrices D(0), D(1) and D(2) used by the Floyd-Warshall algorithm;
c. Considering the graph undirected (remove the orientation of the edges), compute a MST with Kruskal

C. Problem (8p)

(8p) C1. A city has n key objectives and you know all the roads and distances between them. The city municipality wants to build a single emergency point with the condition that it must be the most central of the objectives (the distance to any other objective must be minimum). The number of roads is of the same order as the number of objectives.
Write your idea for solving the problem, the pseudocode, the complexity and the correctness of your solution! Try to find the optimal solution!