Algorithm Design and Complexity – Exam
2014, January 20th
SUBJECT A

TIME: 90 minutes
POINTS: 40 points

A. Theory (20p)

(4p) A1. When we are computing the complexity of an algorithm, we are usually interested in the asymptotical running time in the worst case. Explain what do "asymptotical running time" and "worst case" mean for the complexity of an algorithm.

(4p) A2. Why is it inefficient to use backtracking? Then, why and when should backtracking be used? Write three possible heuristics that may be used to improve backtracking.

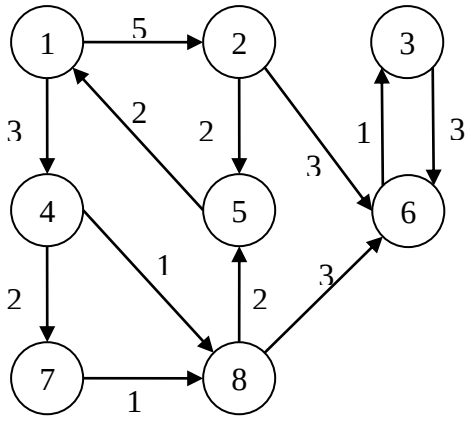
(4p) A3. In Kosaraju's algorithm for computing the strongly connected components, why is it important to choose the roots of the depth trees in a particular order when running DFS on the transpose of the graph G? State what is the order and explain.

(4p) A4. Starting from the pseudocode of Prim's algorithm, determine the general running time. Explain how this running time varies accordingly to the various methods used to implement the priority queue.

(4p) A5. Give an example of a problem that can be solved both by greedy and dynamic programming. Starting from this problem, explain the differences between the two techniques.

B. Exercices (12p)
(6p) B1. Solve the following recurrences (choose two from below, the third one is a bonus exercise):

a. Given a function f: N → R, compute the result of the following expression: $O(f(n)) + \Theta(f(n)) = \ldots(f(n))$
b. $T(n) = 4 * T(n/3) + n^{1.5}$
c. $T(n) = T(n/4) + T(3n/4) + n$



(6p) B2. Considering the graph on the left, show how the following algorithms operate (choose two from below, the third one is a bonus exercise):

a. DFS from node 1, by computing the discovery and finish time for each vertex (ignore the weights for this point);
b. Dijkstra's algorithm from node 5, by pointing out the nodes in the queue (Q) at each step, as well as the values of the distances array d;
c. The first 3 steps of Floyd-Warshall algorithm by computing D(0), D(1), D(2) and D(3)

C. Problem (8p)

(8p) C1. An orchard is modeled as a 2-dimensional matrix A(n, m). Each cell in the matrix contains an apple tree that has A[i][j] kilos of ripen apples. The farmer is located in the upper-left cell of the orchard and can only move either one cell to the right or one to cell down until he reaches the lower-right corner of the orchard. Help the farmer choose the best path of moving through the orchard in order to maximize the quantity of apples that he harvests knowing that he is collecting all the ripen apples along this path.

Write your idea for solving the problem, the pseudocode, the complexity and the correctness of your solution! Try to find the optimal solution!