Write your solutions in this sheet below the problem statement. Mark clearly your solutions by corresponding letters A, B, C, and separate visually each solution from the other ones by empty space or by a line. Use the other side of the sheet or ask for an additional blank sheet if necessary.
Each problem 1. – 4. is worth 0 – 3 points, each answer to a particular question A, B, C contributes at most 1 point to the total.

1. MST algorithms

A. Boruvka's algorithm processes graph G which is a cycle with *n* vertices. What is the asymptotic complexity of Boruvka's algorithm in this particular case? Explain your reasoning and present the solution using O/Θ/Ω notation depending only on value of *n*.

B. Kruskal's algorithm first processes graph G1 which is a cycle with *n* vertices and next it processes graph G2 which is a complete graph on *n* vertices. What are asymptotic complexities of Kruskal's algorithm in this two particular cases? Explain your reasoning and present the solution using O/Θ/Ω notation depending only on value of *n*.

C. A simple modification of Prim's algorithm can process a complete graph with *n* vertices in time proportional to n^2. The modification does not utilize a priority queue. Describe the modification.

2. An alphabet Z = {*a*, *b*} is given. Also, three languages L1, L2, L3 over Z are given by the following prescriptions: L1 = {*a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb*, ... }, L2 = {*ba*, *bab*, *babb*, *babbb*, *babbbb*, *babbbbb*, ... }, L3 = {*bba*, *bbab*, *bbabb*, *bbabbb*, *bbabbbb*, *bbabbbbb*, ... }. Two more languages, K and L, are given.
K = L1.L2.L3 (concatenation of L1, L2 and L3), L = L1 ∪ L2 ∪ L3 (union of L1, L2 and L3).

A. Determine the number of words in K which length is exactly *n* (suppose *n* > 10). Explain your calculations.

B. Draw transition diagram of a finite automaton A over alphabet Z which accepts language L. Automaton A should not contain any ε-transition.

C. Modify automaton A constructed in the answer to question B. The modified automaton detects words of language L in a text over alphabet Z. Write down the modified automaton in the form of table, in which the rows correspond to the states of the automaton and the columns correspond to alphabet characters. Mark the start state and the final states in the table.

3. Search trees

A. Define *black height* of a R-B tree and calculate the maximum possible number of red nodes in a R-B tree which black height is 3 and which contains maximum possible number of keys.

B. A k-d tree T is originally empty. Insert into T the points given below, in the given order, and draw T after insertion of each key. The points are: (1, 1), (3, 2), (2, 1), (0, 2), (2, 3).

C. Suppose that a k-d tree T contains $31 = 2^5–1$ nodes and that T is perfectly balanced. Determine the number of visited nodes during one operation FindMin which is run form the root of the tree. There is an additional parameter of FindMin operation --- the coordinate along which the search space is divided in the root. Solve the problem for all possible values of this parameter.

4. Heaps

A. *Binomial* heap H contains n^2 items. We insert another n^3 items into H, one by one, using operation Insert. What will be the asymptotic complexity of the entire insertion proces? Explain your reasoning.

B. The *amortized* complexity of operation insert in a binomial heap differs from the *asymptotic* complexity of the same operation in a binomial heap. Explain why these two complexities differ from each other and calculate the difference.

C. When operation ExtractMin is applied on a *Fibonacci* heap the Consolidation operation is applied on the heap as well. Suppose that a Fiboacci heap contains n items. Decide and explain whether heap Consolidation operation can run in time asymptotically shorter than $\Theta(\log n)$.