Written Theory Qualifier Exam

Your number: ___________________________

Time limit: 2.5 hours. Use only the notes supplied by the Department
Fall 2019, CS Department, IIT

For every question, please write your answer in a clean and concise way. Use additional pages, start a new page with each problem and write only on one side of the paper.

Use procedures if you want – marking clearly what the parameters are and what they do, and with what running time in terms of its parameters. Unless the procedures are from the textbook, write pseudocode for the procedures. You should be given a copy of this textbook.

Problem 1. Suppose you are given \( k \) \( n \)-element sorted sequences \( A_i \), each representing a set (none has duplicate entries). Describe an \( O(nk \log k) \)-time method for computing a sorted sequence representing the set \( \bigcup_{i=1}^{k} A_i \) (with no duplicates).

Pseudocode is required. You do not have to argue correctness (but, obviously, your method must be correct), but must justify the running time.

Problem 2. This problem refers to the DFS(G) and DFS-Visit(u) procedures from the textbook, applied to an undirected graph. Show how to modify depth-first search so that it assigns to each vertex \( v \) an integer label \( v.cc \) between 1 and \( k \), where \( k \) is the number of connected components of \( G \), such that \( u.cc = v.cc \) if and only if \( u \) and \( v \) are in the same connected component of the input undirected graph.

Write the pseudocode of your modified DFS(G) and DFS-Visit(u). Argue that your modifications do not increase the running time of the algorithm by more than a constant factor.

Problem 3. Suppose you have one machine and a set of \( n \) jobs \( a_1, a_2, \ldots, a_n \) to process on that machine. Each job \( a_j \) has a processing time \( t_j \), a profit \( p_j \), and a deadline \( d_j \). The machine can process only one job at a time, and job \( a_j \) must run uninterruptedly for \( t_j \) consecutive time units. If job \( a_j \) is completed by its deadline \( d_j \), you receive a profit \( p_j \), but if it is completed after its deadline, you receive a profit of 0. Give a polynomial-time algorithm to find the schedule that obtains the maximum amount of profit, assuming that all processing times are integers between 1 and \( n \).

It may be helpful (and it is OK here) to assume that the jobs are sorted by increasing order of deadlines.

Present pseudocode, discuss correctness, and analyze the running time of your algorithm.