

August 2014 QE

Questions 1 and 2 are worth 10 points correct, but -10 points if incorrect. They are worth zero unanswered. No explanations will be considered, and the CE-1 overall score cannot be negative.

In questions 3 and 5, an English algorithm description is requested. In each of these problems, you may provide pseudo-code as well, for clarity, but your English description must be complete and the pseudo-code may be ignored at our discretion.

1. (Answer True or False only) If a problem L_1 is polynomial-time reducible to a problem L_2 and L_1 has a polynomial-time algorithm, then L_2 has a polynomial-time algorithm.
2. (Answer True or False only) Every language with finitely many members is in \mathcal{NP} .
3. (25 points) Describe a linear algorithm for perfectly shuffling a deck of n cards. Assume you are provided a constant-time random-number generator that accepts a positive integer k and uniformly randomly returns a positive integer no larger than k . Assume the input to the algorithm is an array $A[1..n]$ of pointers to cards. Do not use recursion in your algorithm description. Your answer must include three parts:
 - a. An English description of your algorithm. (You may provide pseudo-code for increased clarity, but your English description must be complete and any pseudo-code may be ignored in grading at our discretion.)
 - b. A correctness claim for the algorithm: completely characterize the desired input/output behavior. (You do *not* need to argue the correctness, just completely characterize the goal.)
 - c. For each loop, a loop invariant that holds for the loop (you do *not* need to argue that the invariant holds). This invariant must be strong enough that any loop body that meets the invariant can be substituted for your loop body and still give correct code. (You do *not* need to argue the correctness of the loop invariant.)
4. (25 points) A simple path in a (weighted) graph is a path containing no cycles. Dijkstra's algorithm finds the lowest-weight simple path from a given source to each other vertex. Consider replacing the min-oriented priority queue in Dijkstra's algorithm with a max-oriented priority queue, and replacing the use of minimization in updating distance values with maximization. Initialize all distance values to zero. Does the resulting algorithm find the highest-weight simple path from the source to each other vertex? Justify your answer carefully.
5. (30 points) Consider a directed acyclic graph where the vertices represent tasks and the edges represent pre-requisite relationships. Specifically, the vertices are t_1, \dots, t_n and an edge (t_i, t_j) represents the information that task t_i must be completed before task t_j can be started. Suppose you have a positive real function r on tasks so that $r(t)$ gives the time required for task t to be completed once started. Describe in English an algorithm to compute the minimum time to complete all the tasks in the graph, assuming that arbitrarily many tasks can be underway at one time as long as each task t has all its prerequisite tasks already completed when t is started. Argue for the correctness of your algorithm. State the worst-case run-time complexity of your algorithm and justify your claimed complexity.

Write in Exam Book Only