

Homework 9 – Due Friday, April 20, 2007

Reminder

- Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.

Exercises These should not be handed in, but the material they cover may appear on exams: non-starred exercises in Chapters 22.1, 22.2, 22.3, 22.4, 23.1, 23.2.

Problems to be handed in

1. (**Short Answers**) For each of the following give an answer and a short justification.
 - (a) Consider a directed graph with vertices a, b, c, d, e, f and edges $(a, b), (b, c), (c, f), (a, d), (d, e), (e, f)$. How many topological orderings does this graph have?
 - (b) Recall that on the previous homework, you were asked to design an algorithm to detect whether a given undirected graph has a cycle (and output a cycle if it exists). Argue that it can be done in time $O(n)$, where n is the number of nodes in the graph, no matter how many edges are in the graph.
 - (c) How can we use Prim's algorithm to find a spanning tree of a connected graph with no edge weights?
 - (d) Is the algorithm in part (c) efficient? If not, give a more efficient algorithm and analyze its running time.
2.
 - (a) (**Topological Sort**) Exercise 22.4-1 from CLRS.
 - (b) (**Number of paths in a DAG**) Exercise 22.4-2 from CLRS. Explain why your algorithm runs in time $O(n + m)$.
3.
 - (a) (**Lightest edge is in MST**) Exercise 23.1-1 from CLRS.
 - (b) (**Divide-and-conquer for MST**) Exercise 23.2-8 from CLRS.
4. (**Greedy Scheduling**) Consider the problem of scheduling n jobs of known durations t_1, t_2, \dots, t_n for execution by a single processor. The jobs can be executed in any order, one job at a time. You want to find a schedule that minimizes the total time spent by all the jobs in the system. (The time spent by one job in the system is its finishing time, assuming that the first job

started running at time 0.) Design and analyze a greedy algorithm for this problem. To prove correctness, consider an optimal ordering of jobs. Let $f_1 \leq f_2 \leq \dots \leq f_n$ be the finishing times with respect to the optimal ordering. Prove that the finishing time for the i th job scheduled by your algorithm is at most f_i . Conclude that your algorithm outputs an optimal solution. Do not forget to give the running time of your algorithm.

5. * (**Optional problem**) Exercise 23.1-11 from CLRS. Your algorithm should be much faster than recomputing MST from scratch. Prove correctness and the running time of your algorithm.