CSPP 55001 Algorithms — Autumn 2009

Homework 9 (assigned November 26, due December 2)

Reading: CLRS sections 16.1–16.2; chapter 34

Written assignment: Solve the following "Do" exercises and assigned problems. Only solutions to the assigned problems should be turned in.

Note: You are responsible for the material covered in both "Do" exercises and assigned problems.

Note: If you work with others, indicate their names at the top of your homework paper. Everyone must submit their own independently written solutions.

"Do" Exercises (not to be handed in):

1. Exercise 16.1-2 on CLRS page 422.
2. Exercise 16.2-4 on CLRS pages 427–428.
3. Exercise 16.2-5 on CLRS page 428.
4. Exercise 34.2-2 on CLRS page 1065.
5. Problem 34-1, parts a–c, pages 1101–1102.
   2nd Edition, Problem 34-1, parts a–c, page 1018.

Problems (to be handed in):

Important: When describing an algorithm in pseudocode, explain the meaning of your variables in English. Comment the lines of your pseudocode. Your pseudocode should be short, clear, and complete to receive full credit. If your code is long, difficult to read, or incomplete, you will not receive full credit.

1. Suppose that we have a set of lectures that need to be scheduled and a large number of classrooms. Any lecture can take place in any classroom, but any two lectures that overlap in time must be scheduled in different classrooms. We wish to schedule all the lectures, using as few classrooms as possible. Design an efficient greedy algorithm that schedules all the lectures using the minimum possible number of classrooms. Describe your algorithm in pseudocode. Analyze its running time. Argue that your algorithm is correct. (15 points)

2. Consider a modification to the interval scheduling problem in which each task \( t_i \) has, in addition to a start and finish time, a value \( v_i \). The objective is no longer to maximize the number of tasks scheduled, but instead to maximize the total value of the tasks scheduled. That is, we wish to choose a set \( T \) of nonoverlapping tasks such that the sum of the values \( v_i \) for all \( t_i \in T \) is maximized. Give a polynomial-time algorithm for this problem. Describe your algorithm in pseudocode. Analyze its running time. Argue that your algorithm is correct. (15 points)

3. Given a graph \( G = (V, E) \), a vertex cover of \( G \) is a subset \( S \) of vertices such that for every edge \( (u, v) \)}
in $E$, either $u \in S$ or $v \in S$ or both. A \textit{minimum vertex cover} of $G$ is a vertex cover of $G$ with fewest vertices. Consider the following greedy algorithm for computing a minimum vertex cover of a graph.

1. $S := 0$
2. Pick a vertex $u$ with maximum degree and add it to set $S$
3. Delete $u$ from $G$ along with all edges incident on $u$
4. Stop if $G$ is empty. Otherwise go to step 2.

Exhibit a tree with 10 vertices for which this algorithm does not produce a minimum vertex cover.

(10 points)

Note: The \textit{degree} of a vertex is the number of edges incident on it. If there are several vertices with maximum degree, the algorithm picks any one of these arbitrarily.

4. A \textit{hamiltonian path} in a graph is a simple path that visits every vertex exactly once. Show that the Hamiltonian $(u, v)$-path problem, viz., given a graph $G = (V, E)$ and two vertices $u$ and $v$ in $V$, is there a path starting at $u$ and ending at $v$ that goes through each vertex exactly once, can be solved in polynomial time on directed acyclic graphs. Give an efficient algorithm for the problem, i.e., your algorithm either gives a hamiltonian path or it reports that there is no hamiltonian path. Describe your algorithm in pseudocode. Prove the correctness of your algorithm and analyze its running time. (15 points)