COS 423 Problem Set 6 Due Tuesday May 10, 2011

Spring 2011

Tarjan

Collaboration allowed on Problem 1 but not on Problems 2 and 3. Cite your sources and/or collaborators, and do not share your written solutions; your write-ups should be completely your own.

1. (Routing) Given is a directed graph with *n* vertices, *m* edges, and a source vertex *s*. Source *s* is a warehouse; the other vertices are delivery sites. Each site requires one delivery from the warehouse. The owner of the warehouse has *k* trucks and needs to plan routes for the trucks to make all the deliveries. Each truck will proceed from *s* along a path in the graph, making deliveries at some or all of the vertices it visits, and return to *s* by the end of the day. A day consists of *t* units of time. A truck can traverse an arc or make a delivery in one unit of time.

(a) Prove that it is NP-complete to determine if there is a way to route the trucks to make all the deliveries, even if *k* is fixed. How small can you make *k* and still have the problem be NP-complete?

(b) Now suppose that the graph is strongly connected but every cycle contains *s*, and that *t* > 2*n* – 2. Give a polynomial-time algorithm to find the minimum number of trucks needed to make all the deliveries, and to find routes for the trucks. Hint: reduce this problem to a network flow problem with lower bounds on capacities, in which the goal is to find a flow of minimum value. Transform such a problem into a maximum flow problem by starting with a pseudoflow that satisfies all the lower bounds, and constructing a flow problem whose solution gives a way to convert the pseudoflow into a flow (by converting its excesses and deficits to zero).

2. (Scheduling) Given is one machine and a set of n tasks, each of which has a processing time, a deadline, and a profit. The machine can do one task at a time; each task, once assigned to the machine, must run to completion. If a task is completed by its deadline, one earns the profit associated with the task. The goal is to schedule the tasks on the machine to earn the maximum profit.

(a) State this problem as a decision problem, and prove that the decision problem is NP-complete. (Hint: relate this problem to the subset sum problem.)

(b) Suppose all processing times are integers in the range from 1 to *n*2. Give a polynomial-time algorithm to solve the optimization problem. (Hint: use dynamic programming – for each time slot, consider the possibilities for the task, if any, that finishes at the end of the slot.)

3. (Maximum-residual-capacity augmentation) The purpose of this problem is to analyze and implement the version of the augmenting path method for maximum flow in which each augmenting path has maximum residual capacity. Let *G* be a symmetric directed graph with source vertex *s*, sink vertex *t*, and non-negative arc capacities. Let *f* be a flow on *G* of value *F* and let *f’* be a maximum flow on *G*, of value *F’*.

(a) Prove that the graph of residual arcs with respect to *f* has a flow *f’’* of value *F’* – *F*.

(b) Prove that there is an augmenting path with respect to *f* of residual capacity at least (*F’* – *F*)/*m*. (Hint: Decompose the positive part of *f’’* into at most *m* paths of flow.)

(c) Suppose all arc capacities are integers in the range from 1 to *U*, *U* > 1. Consider the augmenting path method in which each augmentation is on a path of maximum residual capacity. Prove that the number of augmentations is O(*m*lg*U*).

(d) Give a fast algorithm to find an augmenting path of maximum residual capacity and analyze its running time Full credit will be given for an O(*m*)-time, which would give an overall time bound of O(*m*2lg*U*) for finding a maximum flow based on the result in (c). Partial credit will be given for an algorithm whose running time is within a log factor of linear. To get a linear running time, you may assume the existence of a linear-time algorithm for median-finding.)