CS:3330 Mini-Exam 2, Fall 2015

Due: Tue, Dec 8 2015 by class time (9:30 am)

---

1. Consider the following variation on the *Interval Scheduling* problem. You have a processor that can operate 24 hours a day, every day. People submit requests to run *daily jobs* on the processor. Each such job comes with a *start time* and an *end time*; if the job is accepted to run on the processor, it must run continuously, every day for the period between its start and end times. (Note: certain jobs can begin before midnight and end after midnight; this makes for a type of situation different from what we saw in the Interval Scheduling problem.)

   Given a list of n such jobs, your goal is to accept as many jobs as possible (regardless of their length) and subject to the constraint that a job can run at most one job at any given point in time.

   Note: If this problem seems familiar to you, it is because most of you have already attempted it as part of HW4.

   (a) Provide an algorithm to do this with a running time that is polynomial in n. You many assume for simplicity that no two jobs have the same start or end times.

(b) Prove the correctness of your algorithm.

2. Consider the directed shown below. The edges of this graph have weights, but I have hidden them from you. We start Dijkstra's shortest path algorithm with source s and after 2 iterations of the algorithm the set $S = \{s, a, b\}$, the d-values are $d[s] = 0$, $d[a] = 3$, and $d[b] = 7$, and the d'-values are $d'[c] = 12$, $d'[d] = 8$, $d'[e] = 10$. Eventually, when the algorithm completes execution, the d-values of the vertices c, d, and e are $d[c] = 10$, $d[d] = 8$, and $d[e] = 9$.

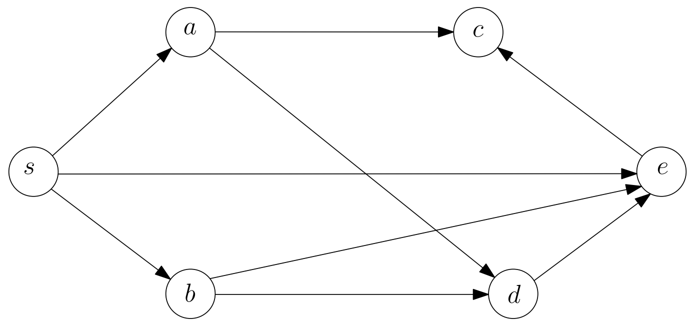

Figure 1: Directed graph with hidden edge weights.

(a) Assign weights to all the edges in the graph such that the above described behavior of Dijkstra's algorithm does indeed occur. Show your answer in the picture above.

(b) Show the shortest paths from s to all the vertices in the graph.

(c) Suppose that we have access to a mysterious data structure that supports the operations `getMin` and `decreaseKey`, running in time $O(\sqrt{n})$ and $O(1)$ respectively on a collection of n items. If we implement Dijkstra's algorithm using this data structure, what would the running time of our algorithm be, as a funtion of n (the number of vertices in the graph) and m (the number of edges in the graph)? Express your answer in asymptotic notation and explain how you obtained your answer.