Problem Set 1—Due Thursday, April 15, 2PM

Homework Info: Homeworks are due by 2PM on the due date. They are to be turned in to me, electronically or in class. No late homeworks will be accepted.

Much of what one learns in this course comes from trying to solve the homework problems, so work hard on them. Doing a conscientious job on the homeworks is the best preparation for the exams. I hope that you will ultimate solve the majority of the problems, but don't be surprised if some of them stump you; the problems may be quite challenging. Note that what you turn is expected to be your own work, not a solution you found online or got from a classmate.

Your solutions should be terse, correct, and legible. Note that when giving a solution algorithm your goal is assumed to be a fast (ideally fastest) solution. Also, you should justify correctness and give its run time.

Understandability of the solution is as necessary as correctness. Expect to lose points if you provide a "correct" solution with a not-so-good writeup. As with an English paper, you can't expect to turn in a first draft: it takes refinement to describe something well. Typeset solutions are always appreciated.

If you can't solve a problem, briefly indicate what you've tried and where the difficulty lies. Don't try and fake it.

(20) Problem 1. In the beam production problem described in class you were given
i) n beam lengths $L_1 < \ldots < L_n$,
ii)the costs F_i for building a factory to produce beams of length L_i,
iii) D_i , the number of beams of length L_i to be produced and
iv) C_i, the cost to produce one beam of length L_i.

For each length L_i we could produce the D_i beams required either by producing a beam of length L_i or by using a larger beam that is cut down to size L_i.

In class we described how to produce a graph such that the shortest path gave us a minimum cost solution (in $\Theta(n^2)$ time). Now consider two variations of this problem. In each case again describe how to formulate and solve this as a shortest path problem. Justify the correctness of your solution method and give its run time.

a) Assume now that there is a cost c to cut a beam to a smaller length (note that c is a fixed cost per beam cut, but it does not depend on the size of the beam cut or produced).

b) Now assume that in addition to the cut cost of part a), we also have a limit L on the amount that can be cut off of a beam (so if we produce a beam of length L_i, we can cut it down to a size of $L_i - L$ or greater only; e.g if $L = 10$ and we produce a beam of length 100, we can only use it for beams of size 90 to 100).

(25) Problem 2. We noted in class that we can find the shortest path from a source to all other vertices in a Directed Acyclic Graph (DAG) with n vertices and m edges in $O(n + m)$ time (see for example, 24.2 in the Cormen, Leiserson, Rivest and Stein Algorithms book).

a) Describe how to efficiently find the **longest** path from a source s to each other vertex in a DAG (note that is problem is NP-hard for general graphs). Give the run time and justify the correctness of your algorithm.

b) Suppose we are given a bound h on the maximum number of arcs we are allowed to traverse. Describe how to efficiently find the shortest path from a source s to each other vertex in a DAG such that each path uses at most h arcs (or report infinity if there is no path using h or fewer arcs).

(15) Problem 3. We discussed the concept of *reduced cost edges* where we have a function $\pi(v)$ on the vertices, and for original edge weights of $w(i,j)$ we convert them to $w_\pi(i,j) = w(i,j) + \pi(j) - \pi(i)$. As we noted (and in the Goldberg-Harrelsonn paper), this shifts the length of all $s-t$ paths by $\pi(t) - \pi(s)$, so we still get the same shortest paths with respect to these new arc lengths regardless of whether the new lengths are all non-negative (feasible) or not. However, we might worry about introducing negative cycles.

Show that for any function π, that if the graph/ original arc lengths had no negative cycles, then the same graph but with reduced cost edges $w_\pi(i,j)$ also has no negative cycles (and thus, if we use an infeasible function π we would introduce new negative arcs, but never new negative cycles).

(20) Problem 4.

Consider bidirectional Dijkstra where we alternate between searches from s and t (but no reduced costs, just use the original ones). Note that in the search from t we reverse the direction of the arcs (i.e. if an arc (a,t) has cost 10 in G, we now use an arc (t,a) of cost 10).

a) Given an example graph where the first vertex scanned in both searches is **not** on the shortest s,t path.

b) Prove that when some vertex is scanned in both directions we have found the shortest path. Specifically, the minimum value $d_s(u) + d_t(u)$ over all vertices u is the shortest path distance from s to t (where $d_s()$ $d_t()$ are the labels in the s and t searches at the point we first scan the same vertex in both).