15-451 Algorithms, Spring 2012

Homework 6 (100 pts) Due: April 9 - 11 oral presentation

Ground rules:

- This is an oral presentation assignment. You should work in groups of two or three. At some point your group should sign up for a 1.5-hour time slot on the signup sheet on the course web page.
- Each person in the group must be able to present every problem. The TA/Professor will select who presents which problem. The other group members may assist the presenter.
- You are not required to hand anything in at your presentation, but you may if you choose. If you do hand something in, it will be taken into consideration (in a non-negative way) in the grading.

Question	Points	Score
1	30	
2	20	
3	25	
4	25	
Total:	100	

1. **Short Questions**

(15) (a) **Preflow Push.** Consider the following undirected flow graph

$$s—v_1—\cdots—v_n—t$$

where all edges have capacity 2 except the edge (v_n, t) which has capacity 1. Show that the Preflow-Push algorithm from class requires $\Theta(n^2)$ relabels and pushes.

(15) (b) **Shortest Path Checking.** Suppose we have a graph $G = (V, E)$ with edge weights w which maybe directed and a start node s. We assume that there are n vertices and m edges in G. Using Dijkstra's algorithm we can compute the distance from s to all other nodes assuming that there are no negative weight edges in G in $O(m \log n)$ time. Show that if you are given the distances from s you can check if these distances are correct in $O(n + m)$ time. Make sure you prove the correctness of your checker.

(20) 2. **Pool construction**

You are working for the International Company for Pool Construction, a construction company which specializes in building swimming pools. A new client wants to build several new pool areas.

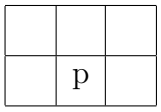
A pool area is a rectangular grid of $w \times h$ square patches, consisting of zero or more (possibly disconnected) pools. A pool consists of one or multiple connected hole patches, which will later be filled with water. In the beginning, you start with a piece of land where each patch is either a hole in the ground or flat grass. In order to transform this land into a pool area, you must adhere to the following:

1. You can leave a patch as it is. This costs nothing.
2. If the patch is grass in the beginning, you can dig a hole there. This costs d.
3. If the patch is a hole in the beginning, you can fill the hole and put grass on top. This costs f.
4. You must place special boundary elements along each edge running between a final grass patch and a final hole patch, to ensure that water does not leak from the pool. This costs b per boundary element.
5. The outermost rows and columns of the pool area must always be grass.

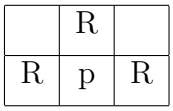
Devise an efficient algorithm for calculating the cost of the cheapest possible pool area given the layout of the existing piece of land.

3. **Castles on a Chess Board**

In this problem you are given an $n \times n$ chessboard with some number of squares occupied, say by your pawns. You wish to place rooks on the board such that no two rooks can attack one another. We ignore the fact that some pawns may attack some rooks. For example, for this board



we can place at most three rooks:



(25) (a) Give an algorithm that maximizes the number of non-attacking rooks by finding a placement in $O(n^6)$ time.

4. **Two New Power Lines**

Using the federal TARP money we want to build two new power lines from New York to LA. A survey has been done on possible routes for these two new power lines and the survey also includes the cost for each segment. The survey consisted of pairs of points in the US with the cost of running a directed power line between these two points. Note that these possible power line segments as well as our final solution the power lines may cross.

More formally we are given a directed graph $G = (V, E)$ with two vertices $s \neq t$ of G and a nonnegative cost function $p : E \to \mathbb{R}$. The goal is to find two edge disjoint paths from s to t of minimum cost.

(15) (a) Show how to find these two edge disjoint paths by using a modified Ford Fulkerson type algorithm. Your algorithm should run in $O(nm)$ time.

(10) (b) Suppose we would like the paths to be vertex disjoint as well except at s and t. How would you modify your algorithm to handle the case of vertex disjoint paths. Your solution should run in the same asymptotic time as before.