Algorithms
Computer Science 140 & Mathematics 168
Instructor: B. Thom
Fall 2004
Homework 11b
Due on Tuesday, 11/16/2004 (beginning of class)

1. [**6 Points**] **Review of Remaining Network Flow Proof.**

   In class, we proved four fundamental theorems which, in the end, allowed us to show that several network flow algorithms were correct. For this assignment, you will re-prove the last one, the Greed Is Good Theorem, **as it was presented in class**.

2. [**10 points**] **Investigating Ones Cuts (Yum)!**

   Suppose someone gave you an original graph G and its residual G_r, and that G_r had no path from s to t. Suggest how you would modify BFS or DFS (you can assume the graphs are both represented as adjacency lists) to:

   (a) Return sets S and T.

   (b) Return the set of edges in the minimum capacity cut.

   Either pseudo-code or a description in English is fine. Also, briefly analyze your approach's runtime. Keep your writeup short. We're looking for big-ideas, not the crossed t's and dotted s's.

3. [**25 points**] **Space Shuttle Profit Optimization!**

   After successful careers at Millisoft, Hurts Car Rental, and the brokerage firm of Weil, Proffet, and Howe, you've been offered two jobs, one by NASA to help them optimize their Space Shuttle program, and the other by Professor Lai to help him get tenure. Your decision, of course, was no Lye.

   On a given mission, NASA will consider a set of experiments that industrial sponsors would like to conduct (for example, "Does gravity affect Spam?"). Let $E = \{E_1, E_2, \ldots, E_m\}$ denote the set of experiments under consideration. Let p_j denote the amount of money the sponsor will pay to conduct experiment E_j. The experiments use a set $I = \{I_1, I_2, \ldots, I_n\}$ of instruments to conduct the experiments. For each experiment E_j, let R_j be the subset of I that contains all of the instruments needed to conduct experiment E_j. Notice that this allows a single instrument to be used in multiple experiments. The instruments are potentially large and heavy and the cost of taking instrument I_k is c_k dollars. Your job is to determine which experiments should be performed in order to maximize the net revenue, which is the total income from the performed experiments minus the total cost of all instruments carried. Amazingly, this problem can be solved using **network flow**!

   We'll construct a network flow problem as follows: The network contains a source vertex, s, vertices $I_1, I_2, \ldots, I_n$, vertices $E_1, E_2, \ldots, E_m$, and a sink vertex t. For each instrument I_k there is a directed edge from s to vertex I_k with capacity c_k (the cost of taking this instrument). For each experiment E_j there is an edge from vertex E_j to t with capacity p_j (the payment for this experiment). Finally, if instrument I_k is in set R_j (meaning that instrument I_k is needed for experiment E_j) then there is a directed edge from vertex I_k to vertex E_j with **infinite** capacity.

(a) First, try this. Consider a situation in which there are three experiments E_1, E_2, E_3 which will bring in 10, 6, and 6 dollars, respectively. Also there are four instruments I_1, I_2, I_3, and I_4 which cost 3, 2, 5, and 7 dollars, respectively to take on the shuttle. Experiment E_1 requires instruments I_1 and I_2, experiment E_2 requires instruments I_1 and I_3, and experiment E_3 requires instruments I_3 and I_4. Using brute-force, enumerate all seven possible combinations of experiments that could be taken. Then, identify the most profitable combination (what experiments it runs, what instruments are required, and what the overall profit is).

(b) For the example problem above, construct the corresponding network flow problem. Find the maximum flow in the network (show your residual graphs at each step) and then find the corresponding cut whose capacity is equal to this flow.

(c) Let's let S denote the vertices that are on the same side of the above cut as vertex s and let T denote the vertices that are on the same side of the cut as t. What do you notice about the instruments and experiments that are in the set T?

(d) Now, let τ be the sum of the payments that NASA would receive for performing all of the possible experiments. In this case, $\tau = 10+6+6 = 22$. From τ, subtract the capacity of the cut you found above. Surprise! What is the most profitable combination of experiments, what instruments are required, and what is the net revenue?

(e) Finally, we're ready to generalize all of this into an efficient algorithm for solving the profit maximization problem in general. Assume that we've set up a network flow problem corresponding to a given set of experiments and instruments. Prove that for any cut with finite total capacity, if an experiment E_j is in T (the side of the cut containing vertex t), then all of the instruments used in this experiment must **also** be in T. Be careful and precise.

In both this and the next item, some students find it useful to use the following notation: let X_T be those vertices in instrument set X that lie in T and X_S be the remaining instruments (which must lie in S). Similarly, let Y_T be those vertices in experiment set Y that lie in T and Y_S be the remaining experiments (which must lie in S).

(f) Now prove that the maximum net revenue that can be achieved is simply the total sum τ of the payments that would be received for performing all of the experiments minus the capacity of the cut that would be found by one of our network-flow algorithms. Be careful and precise.

(g) Now summarize all of this step-by-step, producing a general purpose algorithm for:

i. Finding the maximum net revenue, given a set of experiments, instruments, and the corresponding payments and costs.

ii. Specifying which experiments should be run to obtain this revenue.

You should think about the choices you are making—try and make your approach as fast as you can. Then, derive the worst-case running time of your approach assuming there are m experiments and n instruments (use these variables in your final run-time analysis instead of V and E).

4. [**25 Points**] **Graph Partitioning!**

Given an *undirected* graph G, a *partition* of the graph is a division of the vertices V into two sets A and B such that every vertex is in exactly one of A and B. The only constraint on A and B is that neither set can be empty. The *crossing number* of the partition is the number of edges with one endpoint in A and one endpoint in B. Professor Minal Cutts would like to find an algorithm which finds a partition with the smallest possible crossing number. (Fast graph partitioning algorithms are used in a variety of applications ranging from data clustering to circuit design.)

(a) Describe your algorithm. (You may use existing algorithms to help!)

(b) Explain clearly why your algorithm is correct.

(c) Derive the running time of your algorithm. You should try and make your algorithm as fast as you can (exponential time algorithms are unacceptable!).