Problem Set 6

This problem set is due at 11:59pm on Wednesday, May 8, 2013.

Both exercises and problems should be solved, but only the problems should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered by the exercises.

Mark the top of each sheet with the following: (1) your name, (2) the name of your recitation instructor, and the time your recitation section meets, (3) the question number, (4) the names of any people you worked with on the problem, or “Collaborators: none” if you solved the problem completely alone.

Each problem must be turned in separately to Stellar.

You will often be called upon to “give an algorithm” to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph or sentence should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. A proof of the algorithm’s correctness.
3. An analysis of the algorithm’s running time.
4. If appropriate, a worked example or diagram to show more precisely how your algorithm works.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.

NP-Completeness

Exercise 6-1. Do Exercise 34.2-6 in CLRS on page 1066.

Exercise 6-2. Do Exercise 34.2-8 in CLRS on page 1066.

Exercise 6-3. Do Exercise 34.4-4 in CLRS on page 1086.

Exercise 6-4. Do Exercise 34.5-1 in CLRS on page 1100.

Exercise 6-5. Do Exercise 34.5-3 in CLRS on page 1100.

Approximation Algorithms

Exercise 6-6. Do Exercise 35.1-3 in CLRS on page 1111.

Exercise 6-7. Do Exercise 35.2-3 in CLRS on page 1117.
Problem 6-1. Pluckmin Pilgrimage

In the real-time strategy game Pluckmin 3, Captain Ottomar needs to move his troop of Pluckmin through hazardous terrain full of obstacles and predators. He has determined that when Pluckmin are traveling through predator-infested areas, there is a maximum number of Pluckmin that can safely travel along each edge without getting eaten. In addition to predators, some “lock” nodes have obstacles that require at least one Pluckmin to clear it by approaching it from another path. Ottomar wants to move as many Pluckmin as he can through this terrain.

Ottomar’s most trusted Pluckmin warrior, Steve, recommends formulating this problem in terms of a modified max-flow problem, where a directed graph $G = (V, E)$ represents the terrain the Pluckmin need to navigate through, and the capacity of each edge $c(u, v)$ is the maximum number of Pluckmin that can navigate it safely. However, the following constraints are added:

- You are also given a set of “lock” nodes $L \subseteq V$. Every lock node $v_l \in L$ contains exactly two edges flowing into it, a input edge $(v_i, v_l)$ and a key edge $(v_k, v_l)$, and exactly one edge flowing out of it, an output edge $(v_l, v_o)$.
- Another requirement for a flow $f$ to be valid is that for every lock node $v_l$ with key and output edges $(v_k, v_l)$ and $(v_l, v_o)$ respectively, if $f(v_k, v_l) < 1$, then $f(v_l, v_o)$ must be 0.

Consider the decision version of this problem, where we ask whether there can be a flow of value at least $k$. Show that this problem is NP-Complete.

Problem 6-2. Factory SAT

Knowing that adaptability is a key to successful business, you own factories each of which can be quickly changed to produce one of two different goods. For example, one factory might be able to smelter aluminum or high-carbon steel, but not both. In addition, for maximum diversity, the goods each factory can produce are completely disjoint (that is, if one factory can choose to produce good $a$ or $b$, and another factory can choose to produce good $c$ or $d$, then $a, b, c, d$ are all distinct).

It’s military contracting season again and now you want to decide what to produce for the next year. Every group in the military has their three favorite projects, each of which requires a good. If you can produce goods for one of the three projects for a group, you will win that group’s contract for this year. It isn’t helpful to produce more than one good for a given group, as each group only offers one contract a year. Additionally you may assume that, for each group, each of the three desired goods is a good that exactly one of your factories has the option of producing.

You want to maximize the number of military contracts you will win this year. We will call this the MOST-MICC (Most Military Industrial Complex Contracts) problem.

(a) Prove that the decision version of MOST-MICC (deciding whether you can win at least $k$ contracts) is NP-Hard.

(Hint: reduce from 3-SAT).
(b) Give a deterministic, polynomial-time 2-approximation algorithm for MOST-MICC.

We can do even better with randomization. Consider the following algorithm:

- Assign all of your factories randomly and count how many of the contracts are satisfied.
- If at least a $7/8$ fraction of the contracts are satisfied, then keep this assignment for the year.
- Otherwise repeat (i.e., try another random assignment).

(c) First, prove that there must exist a selection of products which will allow you to win at least a $7/8$ fraction of the contracts available.

(d) Given a positive integer $n$, consider a random variable $q$ which takes on integer values in the range $[0, n]$. If $E[q]$ is also an integer, show that:

$$\Pr \left[ \frac{q}{n} \geq E \left[ \frac{q}{n} \right] \right] \geq \frac{1}{n+1}.$$

(e) Now analyze the expected running time of the algorithm.

**Problem 6-3. War Relief**

You are in charge of organizing the war relief effort in a war-torn country. There are $k$ food supply centers, $\{s_i \mid i = 1, 2, \ldots, k\}$, and $k$ refugee camps, $\{t_i \mid i = 1, 2, \ldots, k\}$, all connected by a network of highways, $G = (V, E)$. For each $i$, you would like to send food from center $s_i$ to camp $t_i$ via a path in $G$. Problematically, these highways are controlled by local warlords, and you do not want to give any warlord too much influence by using his highway too many times. So your plan is to choose one path $p_i$ from $s_i$ to $t_i$ for every $i = 1, 2, \ldots, k$ while minimizing the maximum “influence” of any highway. The influence of a highway $e$ is the number of paths that use it, that is,

$$\text{influence}(e) = \left| \{i \mid e \in p_i\} \right|.$$

(a) Let $P_i$ be the set of all paths in the network connecting $s_i$ to $t_i$. For every $p \in P_i$, introduce a variable $f_{p,i} \in \{0, 1\}$ to represent whether we decide to send food from $s_i$ to $t_i$ along path $p$. Design an integer linear program (ILP) in terms of these variables $f_{p,i}$ to solve the minimization problem stated above.

*(Hint: you might want to introduce one more variable.)*

Since ILP is NP-hard, we do not know how to solve the program in Part (a) directly. One way to get around this problem is to relax the integrality constraints. In our example, we would relax each integer $f_{p,i}$ to a real number in the unit interval, namely, $f_{p,i}^* \in [0, 1]$. This is called the LP-relaxation.

(b) Assume that we solve the LP-relaxation of the ILP in Part (a). Now,
1. interpret the optimal solution, \( \{ f^*_{p,i} \} \), to the LP-relaxation in terms of flows from \( s_i \) to \( t_i \) for \( i = \{1, \ldots, k\} \), and

2. argue that the optimal value of the LP-relaxation is as good or better than the optimal value of the original ILP.

Note that in the solution to the LP-relaxation, each \( f^*_{p,i} \) is a fractional value, so to get a valid solution to our original war relief problem, we need to round each \( f^*_{p,i} \) to an integer \( f'_{p,i} \in \{0, 1\} \). Our hope is that such rounding will not change the value of the LP too much. Note that

\[
\sum_{p \in P_i} f^*_{p,i} = 1.
\]

This motivates the following randomized rounding scheme: independently, for every \( i \in \{1, 2, \ldots, k\} \), we will randomly choose exactly one path \( p \in P_i \) to connect \( s_i \) and \( t_i \) according to the \( f^*_{p,i} \). Specifically, we choose each path \( p \) with probability \( f^*_{p,i} \). Think of this as setting the variable \( f'_{p,i} \) to 1 when we choose path \( p \), which is still a valid solution to the LP-relaxation, and also the original ILP.

(c) Let \( C' \) be the value of the LP-relaxation after the rounding. Show that when \( C^* \) (the optimal value of the LP-relaxation before rounding) is at least \( 6 \log |E| \), then \( C' \leq 2C^* \) with high probability.