15-451 Algorithms, Spring 2010

Homework #5

April 8, 2010

• This is an oral presentation assignment. You should work in groups of three. Please sign up for a 45-minute time slot for your group (link will soon be provided on the webpage). The later you do this, the less likely you are to get a good time slot.

• Groups of three. We might take points off for smaller groups this time. If you cannot find a third member, post on the message board and contact the TAs.

• Sign up for one slot only. In case this wasn’t obvious, it is extremely bad manners to sign up for two slots and skip one, not informing your TA.

• Each person in the group must be able to present every problem. The TA 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.

1 Q1: CalvinBall (30 pts)

Every four years billions of CalvinBall fans eagerly watch their favourite team, hoping they will win a spot in the playoffs and reach the World Cup. However, many teams are mathematically eliminated” before the end of the season. The most obvious case of elimination is when a team cannot win enough games to catch up to the current leader in their division, but sometimes the situation is trickier. For example, consider the following division:

<table>
<thead>
<tr>
<th>Team</th>
<th>Won</th>
<th>Lost</th>
<th>Games Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>35</td>
<td>19</td>
<td>28</td>
</tr>
<tr>
<td>B</td>
<td>29</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>C</td>
<td>31</td>
<td>23</td>
<td>28</td>
</tr>
<tr>
<td>D</td>
<td>23</td>
<td>32</td>
<td>27</td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>46</td>
<td>27</td>
</tr>
</tbody>
</table>

Team E is clearly behind, but the die-hard fans have not given up yet. They claim that if their team wins all 27 of their remaining games, they will end the season with 36 wins, more than any other team has now. So as long as every other team loses every game... but that’s not possible, because some of those other teams still have to play each other. The answer, indeed, depends on which team plays which in the upcoming games.

Our input consists of two arrays: Win of size n, and Games of size n × n, and an integer 1 ≤ k ≤ n. Win[i] is the number of games team i has already won, and Games[i, j] is the number
of upcoming games between teams $i$ and $j$. Note that $Games$ is a symmetric matrix with zeros on the diagonal: $Games[i, i] = 0$ for all $i$, and $Games[i, j] = Games[j, i]$.

Our goal is to determine whether team $k$ can end the season with the most wins (possibly tied with others).

Design an algorithm to solve the problem [Hint: flow]. Prove its correctness and analyze its complexity.

2 Q2: NP, IP, LP (40 pts)

For each of the following problems:

- State the equivalent decision problem, and show how to solve the original problems using reduction to the decision problem.
- For this part only, choose just one of the problems (instead of both). Show that the decision problem is NP-complete for the problem you chose, and justify your answer.
- Reduce the original problem to an equivalent Integer Programming problem, and show 1-1 mapping between the IP solutions and the solutions to the problem.
- Relax the IP problem to a LP problem. Does the LP problem solve the original problem? That is – is the value of the LP solution identical to the max / min value of the original problem? Prove.

1. Bruce Wayne’s Sonar System: Bruce Wayne wishes to safeguard Gotham. Thus, he wish to install his super-sophisticated sonars all over Gotham, so he can observe all $n$ houses in Gotham. If he installs a sonar in house $i$, it will cost him $c_i$, and will allow him to see what’s going on all the houses in the set $S_i$. What is the minimum cost Bruce Wayne can spend in order to install enough sonars to be able to guard each and every house in Gotham?

2. Pairwise Kindey-Exchange:

Thousands of kidney patients in the US alone are registered on a waiting list for a transplant of a kidney. Because healthy people have two kidneys (and can remain healthy on only one), it is possible for a kidney patient to receive a live-donor transplant, as long as there is a biological match.

A ”pairwise-exchange” is a technique of matching willing living donors to compatible recipients. For example a spouse may be more than willing to donate a kidney to their partner but cannot since there is not a biological match. The willing spouse’s kidney is donated to a matching recipient who also has an incompatible but willing spouse. The second donor must match the first recipient to complete the pair exchange. Typically the surgeries are scheduled simultaneously in case one of the donors decides to back out and the couples are kept anonymous from each other until after the transplant. (See http://en.wikipedia.org/wiki/Organ_transplant#Paired-exchange )

Suppose there are $n$ pairs of people, each pair consisting of a donor and a receipient. The donor of pair $i$ is able to donate a kidney to the receipients in pairs $S_i$ (and they, in return, are able to return the favour). Assuming pairs $i$ and $j$ are suitable for an exchange, the chance
of the exchange succeeding is $p_{ij}$. Our goal is to maximize the expected number of exchanges that succeed. What is the maximum you can achieve?

(Hint: In this course you only studied a very limited number of poly-time solvable problems, and even a more limited number of NP-Complete problems. Two of those are reducible from/to these problems.)

3 Q3: Hungry nodes (30 pts)

Consider a modification to the network flow problem where nodes "eat up" some of the flow coming into them. That is, across any node $n$, Flow Out($n$) = max(Flow In($n$) − $e_n$, 0), where $e_n$ is some known constant. We want to find the maximum flow to the sink, i.e. $max$(Flow In($t$) - Flow Out($t$)), in a directed graph $G$ with source $s$ and sink $t$. Consider the following linear program to do this.

Variables. Set up one variable $x_{uv}$ for each edge $(u, v)$.

Objective. Maximize $\Sigma u x_{ut}$

Constraints.

1. $0 \leq x_{uv} \leq c(u, v)$.
2. $\forall v \notin \{s, t\}, \Sigma u x_{vu} \geq \Sigma u x_{uv} - e_v$
3. $\forall v \notin \{s, t\}, \Sigma u x_{vu} \geq 0$

Prove that the linear programming solution is correct or provide a counterexample.

Extra Credit: Give a correct linear programming solution or show that this problem is not solvable using linear programming.