Homework 4

Deadline: 11:59pm, April 5, 2013
Available points: 110. Perfect score: 100.

You will receive 10% extra credit points if you submit your answers as a typeset PDF (preferably using LaTeX, in which case you can also submit electronically your source code). There will be a 5% bonus for typewritten but not typesetted answers. Resources on how to use LaTeX are available on the course's website. **Do not submit Word documents, raw text, etc.** Make sure to generate and submit a PDF if you want to get the extra credit points. In this case you can submit your solutions electronically through `sakai.rutgers.edu`.

If you choose to submit handwritten answers and we are not able to read them, you will not be awarded any points for the part of the solution that is unreadable. Handwritten answer-sheets can be submitted to the instructor in class or to one of the TAs during office hours.

Try to be precise. Have in mind that you are trying to convince a very skeptical reader (and computer scientists are the worst kind...) that your answers are correct.

Each pair of students must write its solutions **independently from other teams**, i.e., without using common notes or worksheets with other students. Each pair of students need to submit only one copy of their solutions. You must indicate at the top of your homework who you worked with. You must also indicate any external sources you have used in the preparation of your solution. **Do not plagiarize online sources and in general make sure you do not violate any of the academic standards of the course, the department or of the university (the standards are available through the course's website)**.

Problem 1 (35 points): A popular practice in the recently settled island of Catan is bartering commodities. For instance, two tons of grain can be swapped for one ton of ore, or three tons of wool can be swapped for one ton of lumber. In the past, these commodity trades were not regulated and were highly varying in terms of the quantity of each resource bartered for another. This often led to heated disagreements between the settlers.

In order to maintain peace on the island and make some profit along the way... the Bank of Catan decided to regulate these trades. It required that each trade should go through the bank and follow swapping ratios $R_{i,j}$ and $R_{j,i}$ for a trade involving commodities i and j depending on the level of productivity of each resource. In particular, a settler providing quantity Q_i of commodity i would receive $R_{i,j} \cdot Q_i$ of commodity j and the settler providing quantity Q_j of commodity j would receive $R_{j,i} \cdot Q_j$ of commodity i. These rates are defined so as to satisfy the following requirement: $R_{i,j} \cdot R_{j,i} < 1$. This means that if one settler started with a commodity i, changed it into commodity j and then converted it back to commodity i, she would end up with a smaller quantity of its initial resource i.

A. Give an efficient algorithm for finding the most advantageous sequence of bank-regulated trades for converting wool into ore, given that there are n resources total. Assume that the rates $R_{i,j}$ and $R_{j,i}$ are known for all the pairs of resources and that settlers are willing to participate in trades so that all the types of trades are possible. Analyze the running time of your algorithm.

B. The Bank of Catan is afraid that there might exist a sequence of commodity trades i_1 to i_2 to ... i_k and back to i_1, so that: $R_{i_1,i_2} \cdot R_{i_2,i_3} \cdot \ldots \cdot R_{i_k,i_1} \geq 1$. This would mean that a settler could start with a quantity Q_{i_1} of a certain commodity i_1 and then through a series of trades acquire a higher quantity of the same commodity. This could affect the reliability of the Bank of Catan and its economy overall as settlers could create profit without any production. Provide an efficient algorithm to the Bank of Catan for detecting such undesirable events given available swapping ratios for the n commodities. Give an efficient algorithm to print out such a sequence if one exists. Analyze the running time of your algorithm.

Problem 2 (25 points): Assume that you have available the departure and arrival location and times for m flights among n cities. You are then asked to create a program that will receive a customer's origin and destination cities (from the n cities for which we have flight data) and output the list of flights that will allow the customer to arrive to the destination as early as possible, subject to giving him at least one hour for each connection. Have in mind that you need to take into account the time it takes to switch between flights.

Give an as efficient as possible algorithm to solve the problem. Argue about its running time and prove its correctness.

Problem 3 (50 points): Consider that you have n candidates in elections for k positions in a student governing body. We would like to construct a selection scheme that will return a single winner given that the voters submit ballots that provide their preferences over all the candidates, i.e., a list of the candidates in order of preference. Assume ties are not allowed in the choices of the student voters.
For each pair of candidates A and B it is possible to compute the "preference values" $d[A,B]$ (the number of voters that preferred A over B) and $d[B,A]$ (the number of voters that preferred B over A). Obviously if $d[A,B] > d[B,A]$, then candidate A is preferred over candidate B. Given the preference values for all pairs among n candidates, however, it may not always be possible to pick a winner. This is because majority preferences may look like rock/paper/scissors situations: for each candidate, there can be another that is preferred by some majority.

To address this issue, the following scheme has been proposed, which always provides a winner and an actual ordering of all the candidates given the available ballots. For each pair of candidates A and B it is possible to identify a "preference sequence" $s[A,B]$ of candidates that might imply that A is preferred over B. Such a sequence of size m has the following properties:

- $s_1 = A$ and $s_m = B$
- $\forall\ i \in [1, m-1] : d[s_i, s_{i+1}] > d[s_{i+1}, s_i]$

The "strength" of a preference sequence is defined as the minimum "preference value" along the sequence:

$$t(s[A,B]) = \min_{\forall i \in [1,m-1]} d[s_i, s_{i+1}].$$

The "strongest preference sequence" $s^*[A,B]$ is the preference sequence between candidates A and B that has the maximum strength:

$$s^*[A,B] = \underset{\forall s[A,B]}{\operatorname{argmax}}\, t(s[A,B]).$$

The strength of the strongest preference sequence between two candidates will be denoted as $t^*[A,B]$. If there is no preference sequence between candidates A and B, then: $t^*[A,B] = 0$. Given the above definitions, a new way to provide preference between candidates is the following: Candidate X is preferred over candidate Y if and only if $t^*[X,Y] > t^*[Y,X]$. It can be proven that:

- a transitive relation always arises between the strengths of the strongest preference sequences (i.e., if $t^*[X,Y] > t^*[Y,X]$ and $t^*[Y,Z] > t^*[Z,Y]$, then $t^*[X,Z] > t^*[Z,X]$),
- and there is always at least one candidate that has higher strength of preference over every other candidate (i.e., it is the a winner).

Then, the algorithm is to compute the strength of the strongest preference sequence for every pair of candidates and rank the candidates by finding first the winner, then removing it from the list, finding the next winner, etc.

A. Provide an efficient algorithm for computing the strengths of the strongest preference sequences for every pair of candidates. In this process, describe the underlying data structure needed to compute these values. What is the running time of your solution?

B. Assume that there are 5 candidates: Katrina, Bob, Peter, Miranda and Dana. Furthermore, there are 45 voters who voted as follows:

- 7 voters submitted: Bob > Katrina > Miranda > Dana > Peter
- 8 voters submitted: Miranda > Dana > Peter > Bob > Katrina
- 5 voters submitted: Peter > Katrina > Dana > Miranda > Bob
- 5 voters submitted: Peter > Bob > Miranda > Katrina > Dana
- 7 voters submitted: Katrina > Peter > Miranda > Dana > Bob
- 2 voters submitted: Katrina > Dana > Peter > Bob > Miranda
- 8 voters submitted: Dana > Miranda > Bob > Peter > Katrina
- 3 voters submitted: Katrina > Peter > Dana > Miranda > Bob

Compute all the preference values. Does an undisputed winner arise in this case and why yes or no?

Compute the strengths of the strongest preference sequences for all pairs of candidates with the algorithm you proposed.

Provide the final ranking according to the strengths of the strongest preference sequences.