Math 160 (De Loera) **Homework 3**
Due date: May 23, 2016

INSTRUCTIONS

All homeworks will have many problems, both theoretical and practical.

Programming exercises need to be submitted via SMARTSITE using the assignment boxes NOT DROPBOX.

Other methods of submission without prior approval will receive zero points.

Write legibly preferably using word processing if your hand-writing is unclear. Be organized and use the notation appropriately. Show your work on every problem. Correct answers with no support work will not receive full credit.

1. CHALLENGE: PROBLEMS ON GRAPHS & NETWORKS: We saw how the networks can be used to plan for building projects or to schedule the parts of a project. Here are some some more challenges:

   - Let G be a directed graph with distances or costs on its arcs and two special vertices s, t. We are interested on the "longest" paths using two possible definitions
     - If we call the length of a path the sum of the lengths on the path. Write a model to compute the longest path on a network.
     - If we instead call the length of a path the largest length among arcs the path, write another model to compute the longest path on a network under this definition.
   - Consider again, the TSP for n cities and cost of travel c_{ij}. Write a new model for the TSP, different from the one we saw in class. This time us the binary variables $x_{i,j,k}$, which are 1if the k-th leg of the trip, the salesman goes from city i to city j. Your formulation must have a cubic number of constraints.

2. CHALLENGE: FINDING A GOOD MATCH: Matching problems appear everywhere in applications. E.g., how do students get matched to medical schools? How do workers get assigned to jobs at work? How do organ donors get match to patients in need?

   We can model matchings in graphs. A set M of the edges E is called a *matching* of G if and only if:
   $$\forall u \in V,\ |\{e \in M \,|\, u \in e\}| \leq 1$$
   In other words, a vertex is incident to at most one edge of M. A matching M is said to be *maximal* if there is no matching M' with $M \subset M'$. A maximal matching of the largest cardinality is a *maximum* matching. The size of a maximum matching is denoted by $\texttt{OPT}_G$.

   In the following exercises we will explore several approaches to find a maximum matching in a graph:

   a. Let M be a matching and let us denote by $V(M)$ the set of endpoints of edges in M:
   $$V(M) := \{u \,|\, \exists e \in M,\ u \in e\}$$
   Show that when M is a maximal matching of G, $V(M)$ is a vertex cover of G.

   b. Let M be a maximal matching of G, show that $|V(M)| \leq 2\texttt{OPT}_G$.

c. Now consider the following *greedy* algorithm for finding a maximum matching: Start with an arbitrary edge as the very first matching. Find another edge that does not have a vertex in common with the current matching. If one exists add it to the current matching. Repeat until no more edges can be added.

d. What is the running time of this algorithm on a graph with n nodes and m edges?

e. Give an example where the algorithm fails to find a maximum matching. But show that the greedy method always yields a solution that has at least half as many edges as a true maximum matching.

f. Give now an integer-optimization model (i.e., now based on linear equations, inequalities) to construct a maximum matching of any graph. G.

3. Let G be a graph with two distinguished vertices s, t. An even st-path is a path from s to t with an even number of edges. Show that we can formulate the problem of finding an even st-path with as few edges as possible as a minimum cost perfect matching problem. HINT: Construct an auxiliary graph H from G, make a copy of the graph G, and remove vertices s, t. Call that new graph G'. Construct H starting with the union of G, G' and joining every vertex $v \in G$ different from s, t with its copy in G'. Use H.

4. THEORY PROBLEMS: GEOMETRY OF INTEGER SOLUTIONS:

- MEDITATE ABOUT THIS: In the previous problems you dealt with several types of combinatorial optimization problems. Which ones have polynomial time solution? Which ones are NP-hard? Discuss whatever information you find about their complexity.
- Suppose $S = \{(y_1, y_2) \in Z^2 : y_1 - y_2 \leq 2,\ 3y_1 + y_2 \leq 21, y_1 + 5y_2 \leq 34\}$. Find (a) An inequality description of convex hull of S. (b) Find the extreme points of $conv(S)$.
- Use the branch-and-bound method to solve the following optimization problem. Show the solution graphically:

$$\begin{gathered}\max\ y_1 + 3y_2 \\ \textbf{subject to:}\quad y_1 + 5y_2 \leq 12, \\ y1 + 2y_2 \leq 8, \\ y_1, y_2 \geq 0 \texttt{ integer}\end{gathered}$$

.

- Generate a valid inequality using Chvátal-Gomory cut procedure for the following problem:

$$\begin{gathered}\max\quad y_1 + y_2 + y_3 \\ \textbf{subject to:}\quad 3y_1 + 5y_2 - y_3 \leq 12, \\ y1 + y_3 \leq 7, \\ y1 - y_2 + 2y_3 \leq 9, \\ y_1, y_2, y_3 \geq 0 \texttt{ integer}\end{gathered}$$

.

5. CHALLENGE: Applications to genetics, DNA SEQUENCE ALIGNMENT:

   In this problem we study the problem of DNA sequence alignment. The input is a pair of DNA sequences (similar versions exist when trying to align multiple DNA sequences but we omit this here):

   1: `TTGATCAATGG`
   2: `ATCATACAAGGA`
   and the goal is to understand whether or not they have the same biological function. For this we find the best way to align the sequences to each other. If after alignment, the sequences look "similar enough", we will conclude that they have the same biological function.

   More precisely, the sequences are aligned by introducing gaps. For the above two sequences, a possible way to introduce gaps is as follows (gaps are represented by hyphens):

   1: `TTGAT-CAATGG-`
   2: `ATCATACAA-GGA`
   After introducing the gaps, the sequences are compared by counting the number of mismatches, *i.e.,* the number of locations where the nucleotides differ. For the above example, there are two mismatches: one at location 0, and one at location 2 (using the usual array indexing convention in programming!).

   Let g denote the number of gaps and m denote the number of mismatches in a given alignment, the overall cost c of the alignment is then defined by:

$$c := 2 \cdot g + m$$

   The above alignment has cost $2 \cdot 3 + 2 = 8$. Another possible alignment is:

   1: `TTGAT-CAATGG`
   2: `ATCATACAAGGA`
   which has cost $2 \cdot 1 + 4 = 6$, which is minimal among all possible alignments of these two sequences.

   Let us denote by s (resp. t) the first (resp. second) sequence. We define $c(s, t)$ to be the cost of the alignment of minimal cost among all possible alignments. Finally, we denote by n (resp. m) the length of s (resp. t).

   a. For a sequence s, $s[i : j]$ will denote the substring of s ranging from index i inclusive to j exclusive. Give a formula to compute $c(s[0 : i], t[0 : j])$ as a function of $c(s[0 : i], t[0 : j - 1])$, $c(s[0 : i - 1], t[0 : j])$ and $c(s[0 : i - 1], t[0 : j - 1])$. Think recursively.
   b. Using part a. design and describe an algorithm to compute $c(s, t)$ [**hint:** think dynamically! It is very useful to think of this problem as a type of matching problem].
   c. Implement the algorithm you designed above. A test dataset is available at `https://www.math.ucdavis.edu/~deloera/TEACHING/MATH160/dna.txt`. Each line of the datafile is the list of the first 10,000 base pairs of the genome of the well-known *Escherichia coli* bacteria (why is it famous? Do you know?). There are only two lines corresponding to two different species of this bacteria.
   d. Write code (using MATLAB and/or SCIP) which reads the datafile and outputs the cost of the optimal alignment of the two DNA sequences. Do not forget to Submit the code you wrote. Using C or Python to help yourself is allowed.
   e. Two DNA sequences are considered to have the same biological function if the cost of the optimal alignment, divided by the length of the sequence is smaller than 5%. Do the two DNA sequences in the datafile have the same biological function?