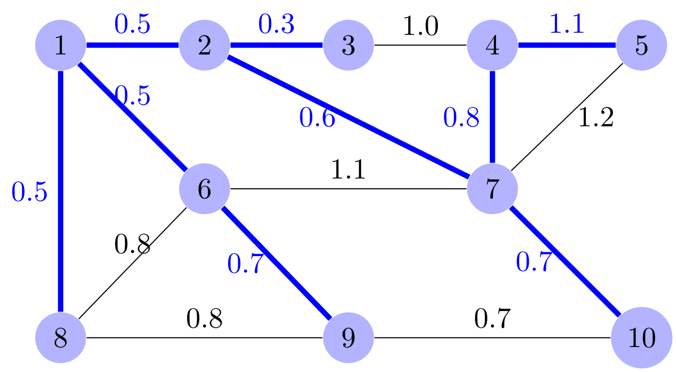
CSCI 3104-Spring 2016: Assignment #9.

Assigned date: Wednesday, 4/17/2016,
Due date: Thursday, 4/28/2016, before class
Maximum Points: 40 points (includes 5 points for legibility).

Note: This assignment *must be turned in on paper, before class.* Please do not email: it is very hard for us to keep track of email submissions. Further instructions are on the moodle page.

P1 (20 points) Let G be an undirected graph with n nodes and m edges. Suppose we have a spanning tree T of the graph that is *claimed* to be a minimum spanning tree. This assignment tries to verify this claim.

As an example, take the graph below with the spanning tree edges shown in blue and edges not in the tree shown in black.



(A) Describe an algorithm to check that T is a spanning tree of the graph (do not check for minimality of T yet).

(B) Let $e : (u, v)$ be an edge of weight w in graph G that *does not belong* to the tree T. As an example, let $e : (9, 10)$ with weight 0.7.

Consider the unique path $u \rightsquigarrow v$ in the tree T. Prove that if any edge in the path has weight strictly greater than w, then T cannot be a minimum spanning tree of G.

(C) For each off-tree edge (u, v) in the example, write down the path $u \rightsquigarrow v$ to check that T is indeed an MST of the graph.

(D) Let e_{min} be the smallest weight edge in the graph G whose edge weights are all different from each other. Use the result from **(B)** to prove that e_{min} must be part of any MST of the graph.

(E) Let G be a graph all of whose edges have different edge weights. Let u be a node in the graph and let (u, v) be the minimum weight edge among all edges that are incident on u. Show that there exists a MST that has (u, v) in it. (**Hint:** Suppose we start Prim's algorithm from the node u as the starting node?)

(F) Write down all the other MSTs of the graph G in the example above.

P2 (20 points) There are five people $P_1, \ldots, P_5$ in a dorm and they share the chores between themselves. There are 5 chores to be done each week: $C_1, \ldots, C_5$. The constraints are as follows:

- Each chore C_i takes time t_i.
- Each person ranks the list of chores from highest preference 1 to least preference 5.
- Each chore is assigned to exactly one person. But each person may get to do multipe chores.
- Let $T = t_1 + \ldots + t_5$ be the total time taken by the chores. No person should do chores that are worth more than 40% of the time. I.e, the the total time taken by the chores for each person cannot be more than $\frac{2T}{5}$.
- The *annoyance* of each person is measured by the sum of the weights of the chores they get to do. For instance if P_2 does a chore that he ranks 4 and another chore that he ranks 1, he obtains an annoyance score A_2 of $1 + 4 = 5$.

Your goal is to write and solve an integer linear programming problem for optimally assigning chores to people. Consider the problem data below:
Time t_i for each chore C_i:

C_1	C_2	C_3	C_4	C_5
5	10	2	8	10

Total time $T = 35$.
Person P_i ranked preferences:

Person	1	2	3	4	5
P_1	C_2	C_1	C_3	C_4	C_5
P_2	C_4	C_2	C_1	C_3	C_5
P_3	C_5	C_2	C_3	C_1	C_4
P_4	C_1	C_2	C_3	C_5	C_4
P_5	C_4	C_3	C_1	C_2	C_5

Let $x_{i,j}$ be a variable that is assigned to 1 if person i does chore j and 0 otherwise.

(A). How many decision variables does the problem have? How do you express that each decision variable can either be 0 or 1?

(B). Write down a constraint for chore C_3 that says that C_3 can be assigned to exactly one person.

(C). Write down a constraint for person P_4 that says that the person cannot work more than $\frac{2T}{5}$ time in total?

(D). Write down an expression for the annoyance A_3 of person P_3.

(E). Write down the integer linear program to minimize the total annoyance $A_1 + A_2 + A_3 + A_4 + A_5$. Solve it and write down the assignment that you obtain.