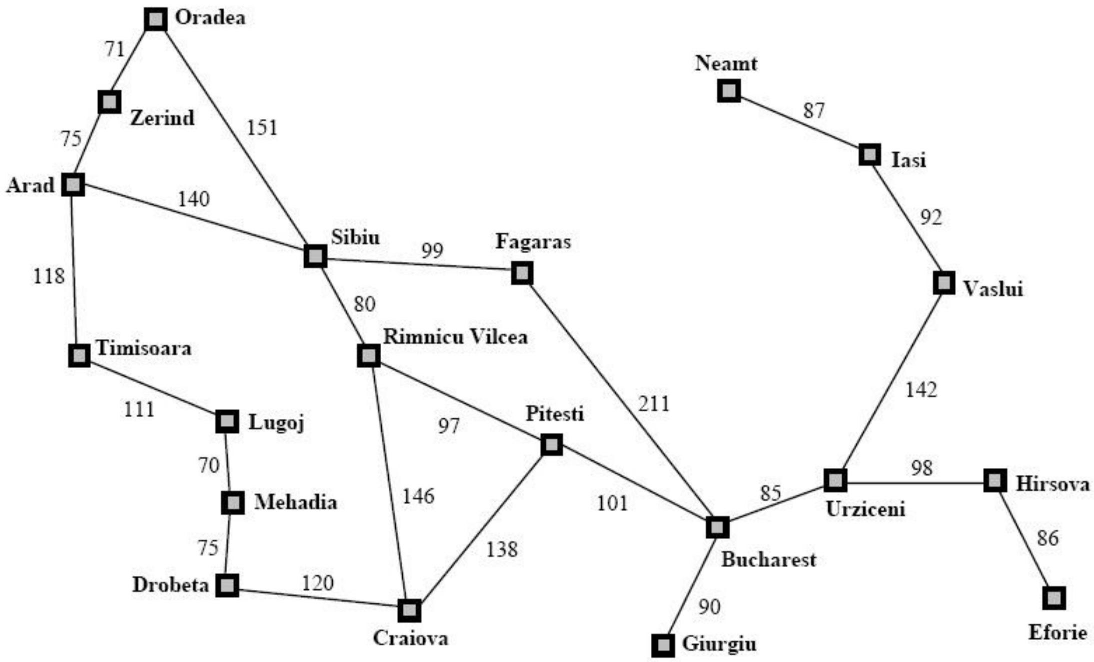
Homework 6

(Due: 05/02/2019 1:45pm)

1. **(3 points)** The following is a map of cities in Romania. Find the minimum spanning tree that connects every city.



2. **(4 points)** Given a connected, undirected graph $G = (V, E)$ where the weights of the edges are either 0 or 1. Show how to compute the weight of a minimum spanning tree (MST) in $O(V + E)$ time.

3. **(3 points) HARD** Suppose that we have a sequence $a_1, a_2, \ldots, a_n$ that we wish to memorize. We memorize it by computing partial sums of the forms $a_i + \ldots + a_j$. Let us say that the **cost** of memorizing a particular integer a is $|a|$. The goal is to use the minimum cost to remember the partial sums so that we can reconstruct the original seqeunce. For example, if $(a_1, a_2, a_3, a_4) = (1, -3, 2, 4)$, one of the possible solution is to memorize:

   - $a_1 = 1$
   - $a_2 + a_3 = -1$
   - $a_2 + a_3 + a_4 = 3$
   - $a_3 = 2$

   The total cost is $|1| + |-1| + |3| + |2| = 7$, and a_1, a_2, a_3, a_4 can be reconstructed by using Gaussian elimination from the information memorized. Note that the minimum cost to memorize the sequence above is 6.

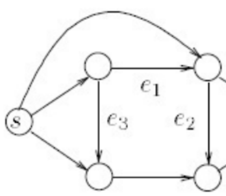
   Give an algorithm that computes the minimum cost to memorize a sequence $a_1, a_2, \ldots, a_n$. The running time of the algorithm should be polynomial in n. (*Hint: MST.*).

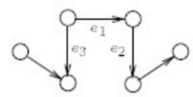
4. **(4 points)** Give an example where Dijkstra's algorithm does not compute the answers correctly when there are negative weight edges.

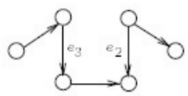
5. **(4 points)** Given a directed, weighted graph $G = (V, E)$ and a weight function $w : E \to [0, 1]$ (that is, the weights of all the edges are between 0 and 1). Suppose that you are initially at vertex s and you hope to escape to an exit at vertex t. The weight of an edge $w(e)$ denotes the chance of survival to travel along e. Given a path $P = (u_0, u_1, \dots u_k)$, the chance of survival to travel along the path P is $\prod_{i=1}^{k} w(u_{i-1}, u_i)$. Show how to compute the path from s to t with the maximum survival probability in $O(E \log V)$ time.

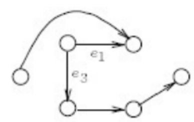
6. **(3 points)** Given a **directed**, **unweighted** graph $G = (V, E)$, a start vertex s, and an ending vertex t. Show how to find two *edge-disjoint paths* from s to t in G in $O(V + E)$ time if they exist. Two paths are edge-disjoint if they do not share any edges.

7. **(4 points)** Show that in the following graph, if we first augment along the path that goes from s to the head of e_3 through e_3 to t and then repeatly augment along the path sequence p_1, p_2, p_1, and p_3, the Ford-Fulkerson will not terminate. Edges e_1, e_2, and e_3 have capacity 1, $\frac{\sqrt{5}-1}{2}$, and 1 respectively. All other edges have capacity 4. (Hint: $(\frac{\sqrt{5}-1}{2})^n = (\frac{\sqrt{5}-1}{2})^{n+1} + (\frac{\sqrt{5}-1}{2})^{n+2}$ for $n \geq 0$.)



p_1 :


p_2 :


p_3 :


8. **(4 points)** Given a graph with n vertices and m edges. The edges are either directed or undirected. Give an algorithm to determine if it possible to orient the **undirected edges** into directed ones (in one of the two directions), such that for each vertex, the number of incoming edges equals to the number of outgoing edges. Your algorithm has to run in polynomial time.