For all questions which involve providing algorithms, prove correctness of the algorithm and provide and justify the runtime of the algorithm.

1 Part A [50 points]

1. Given a connected undirected graph $G = (V, E)$ such that $|E| > n$, and $w : E \rightarrow (1, \infty)$ be a one-to-one weight function on the edges of $G$. (i.e., $w(e) > 1$ for every $e \in E$; and $w(e) \neq w(e')$, for every $e, e' \in E$ such that $e \neq e'$). In this framework, the weights of the edges are determined by querying $w$ on the respective edges. By defining different weight functions, we can ask questions about $G$ with respect to these different weight functions.

Let $w' = w(e) - 1$ and let $w''(e) = w(e) / 2$, for every $e \in E$.

(a) Prove that there exists exactly one MST of $G(V, E)$ with respect to $w$; exactly one MST of $G(V, E)$ with respect to $w'$; and exactly one MST of $G(V, E)$ with respect to $w''$. [5 points]

(b) Let $T_w, T_{w'}, T_{w''} \subseteq E$ be the MSTs of $G(V, E)$ with respect to $w, w'$ and $w''$. Decide whether each of the following statements are correct. Give a proof if it is true, or a counter example if it is not. [5 points]

i. $T_{w'} = T_{w''}$. [5 points]

ii. The minimum cost edge $e_{\text{min}}$ belongs to $T_w$ and the maximum cost edge $e_{\text{max}}$ does not belong to $T_w$. [5 points]

(c) We are given two nodes $u, v \in V$. Assume that there exists exactly one shortest path from $u$ to $v$ in $G$ with respect to $w$; exactly one shortest path from $u$ to $v$ in $G$ with respect to $w'$; and exactly one shortest path from $u$ to $v$ in $G$ with respect to $w''$. Let $P_w(u, v), P_{w'}(u, v), P_{w''}(u, v) \subseteq E$ be these shortest paths from $u$ to $v$ with respect to $w, w'$ and $w''$. Decide whether each of the following statements are correct. Give a proof if it is true, or a counter example if it is not.

i. $P_w(u, v) = P_{w'}(u, v)$. [5 points]

ii. $P_w(u, v) = P_{w''}(u, v)$. [5 points]

2. Let $G = (V, E)$ be a connected undirected graph, with a weight function $w : E \rightarrow \{1, 2\}$. (In other words, all edges of $G$ have weight 1 or 2) Give an efficient $O(|E|)$ algorithm that computes the shortest paths from a single source vertex $s \in V$ to all of the other vertices in the graph $G(V, E)$. [10 points]
3. Given a set of numbers, its median, informally, is the “halfway point” of the set. When the set’s size $n$ is odd, the median is unique, occurring at the $i$th value, where $i = (n + 1)/2$. When $n$ is even, there are two medians, occurring at $i = n/2$ and $i = n/2 + 1$, which are called the “lower median” and “upper median”, respectively. For simplicity in this question, we use the phrase ”the median” to refer to the lower median.

Design and implement a data structure $\mathcal{D}$ to maintain a set of positive integers that supports $Build$, $Insert$, $Extract_X$ and $Plot_X$ (for $X \in \{\text{min}, \text{max}, \text{median}\}$) operations, defined as follows [15 points]:

- $Build_{\mathcal{D}}(S)$: Produces, in linear time, a data structure $\mathcal{D}$ for the set $S$ from an unordered input array. (For implementing $Build_{\mathcal{D}}(S)$, you can use Procedure $Find-Med(S)$, which finds the median of $S$ in linear time.)
- $Insert(\mathcal{D}, x)$: insert element $x$ into $\mathcal{D}$ in $O(\log n)$ time.
- $Plot_{\text{min}}(\mathcal{D})$, $Plot_{\text{max}}(\mathcal{D})$, $Plot_{\text{median}}(\mathcal{D})$: Returns, in $O(1)$ time, the value of the minimum, maximum and the median of $\mathcal{D}$, respectively.
- $Extract_{\text{min}}(\mathcal{D})$, $Extract_{\text{max}}(\mathcal{D})$, $Extract_{\text{median}}(\mathcal{D})$: Remove and return, in $O(\log n)$ time, the value of the minimum, maximum and the median of $\mathcal{D}$, respectively. (Note that you will be removing and returning exactly one of these, depending on which of the three parameters $Extract$ was called with.)
2 Part B (programming, 50 points)

1. Preliminary Problem [10 points]

Suppose we have a graph $G$, and $T$ is a spanning tree (not necessarily with minimum total weight) of $G$. Consider the following operation $SWAP(T, e_1, e_2)$, where we remove $e_1$ from $T$ (conditioned that it is already in $T$) and add $e_2$ to $T$. If the resulting graph $T' = SWAP(T, e_1, e_2)$ is also a spanning tree of $G$, then we will call this a valid swapping.

Show that for any pair of spanning trees $T, T'$ of $G$, it is possible to transform $T$ into $T'$ by a sequence of valid swapping operations. (Hint: Show by mathematical induction that this is possible if $T$ and $T'$ differ by $k$ edges)

2. Programming Problem [30 points]

You are a U.S. senator on a committee for figuring out the best way to install power grid infrastructure within your community. Currently, there are $N$ distribution centers that have been constructed. It has been decided that all of these distribution centers must be connected in a network set up. In order to construct the power line connections, the work must be outsourced to private companies.

There are two main power mainatanance companies that will construct such connections: Maverick, and Desperado. You have approached each one with information regarding your distribution centers, and an amount you are willing to pay for each one, which is constant independent of what distribution centers you are trying to connect. In response, both companies have responded with a list of connections that they are willing to build given your proposed price.

Since each contract has the same price, you will minimize your costs by minimizing the number of contracts you give out. A set of contracts is valid if and only if it satisfies the following:

- All distribution centers are connected within the power grid. There can’t be isolated components, and there should exist a path between every pair of centers.
- The total number of contracts given out is minimized.

Given their responses, you believe it should be relatively straight forward to figure out how to connect all the distribution centers for minimum cost. However, there is a problem... 2016 is approaching, and you forgot that you are also running for president. Part of your campaign image is that you are a down to earth american who enjoys college ball and healthy competition in a capitalist economy. If you gave out too many of the contracts to a single company, then those pesky data scientists with their big data algorithms will point out this fact and use it to attack you for inconsistency.

Suppose that you give $A$ contracts out to Maverick, and $B$ contracts out to Desperado. Then the bias of your assignments will be the absolute value difference of $A$ and $B$. 


Your updated goal is to figure out which subset of contracts to give out, so that all of the distribution centers are connected, costs are minimized, and the bias of your contract selection is minimized.

Write a program to take the contract information, and determine how small of a bias you can achieve for a valid set of contracts. You do not need to determine the actual set of contracts to be outputted.

**Input Format:**
Line 1: \( N \ C \).
The first line indicates there are \( N \) distribution centers labeled 1 through \( N \), and there are \( C \) contracts to consider

Next \( C \) lines: \( i \ j \ c \)
Each of the following lines indicates a contract where \( i \) and \( j \) are distribution center labels, and \( c \) is either "MAVERICK" or "DESPERADO", the name of the company willing to purchase this contract.

The set of contracts will always be enough to yield at least one valid set of contracts.

**Output Format:**
Line 1: \( V \)
Indicating \( V \) is the minimum possible bias for a valid set of contracts.

**Sample Input:**
3 3
1 2 MAVERICK
2 3 MAVERICK
1 3 DESPERADO

**Sample Output:**
0

**Explanation:**
For 3 nodes, you need at least 2 contracts to connect them all. It is possible to select the following contracts:
1 2 MAVERICK
1 3 DESPERADO

Then all of the distribution centers will be connected, while the bias of the contract set is \(-1 - 1- = 0\).

3. **Analysis of programming problem algorithm** [10 points] Describe and give a proof of correctness for the algorithm you used to solve the programming assignment. (Hint: Use the results from the first problem of Part B)

4. **Finding the tree** [Bonus: 15 points] For the programming assignment, we only asked that you compute the value of the minimum possible bias, instead of finding a set
of contracts that achieves it. Give and analyze an algorithm which finds and returns a set of valid contracts which achieve the minimum possible bias.