January 2023 CSE 208
Offline Assignment 1: Minimum Spanning Tree

In this assignment, you need to implement the Prim-Jarnik's algorithm in $\mathcal{O}(|E|\log|V|)$ and Kruskal's algorithm in $\mathcal{O}(|E|\log|E|)$ to find the Minimum Spanning Tree (MST) of a given connected weighted undirected graph.

Input

Take input from a file. The first line will contain two integers n and m, denoting the number of vertices and the number of edges respectively. In each of the following m lines, there will be two integers and a real number u, v, d such that there is an edge (u, v) in the input graph with weight d. Here $0 \leq u, v < n$.

Output

You will implement the two algorithms in seperate files. In either case, print the weight of the MST and the edges in arbitrary order. For Prim-Jarnik's algorithm, mention the root node as well. Save the files as `prim.txt` and `kruskal.txt` respectively. You might also print to the console. If there are multiple MSTs, find any one of them. Check the provided test cases for further clarification.

Submission Guideline

1. Create a directory with your 7 digit student id as its name

2. Put the source files only into the directory created in step 1

3. Zip the directory (compress in .zip format; .rar, .7z or any other format is not acceptable)

4. Upload the .zip file on moodle.

For example, if your student id is 2005xxx, create a directory named 2005xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 2005xxx. Compress 2005xxx into 2005xxx.zip and upload the 2005xxx.zip on moodle.

Sample I/O

Input File

```
7 16
0 2 12
0 3 13
0 4 5
0 5 14
0 6 6
1 2 7
1 3 9
1 4 5
1 5 7
1 6 9
2 4 11
2 5 5
2 6 8
3 4 7
3 5 13
3 6 6
```

Output File `prim.txt`

```
Total weight = 34
Root node = 0
4 1
5 2
6 3
0 4
1 5
0 6
```

Output File `krusal.txt`

```
Kruskal's Algorithm:
Total weight = 34
0 4
1 4
2 5
0 6
3 6
1 2
```

Submission Deadline: Thursday, June 15, 2023, 11:55 PM

Special Instructions

When writing code, it is essential to ensure readability, reusability, and good structure. This involves using appropriate functions to implement algorithms, giving variables meaningful names, adding suitable comments when necessary, and maintaining proper indentation. You will need to use your offline implementation to solve the onlines. There will be a viva too. So, please understand the concepts before you proceed to code.

Please note that you must rely on your own implementation to solve the assigned tasks. It is strictly prohibited to copy code from any source, including friends, seniors, or the internet. **Any form of plagiarism, regardless of its origin or destination, will result in a deduction of 100% marks for the offline assessment**. Moreover, repeated instances of plagiarism may lead to stricter consequences in accordance with departmental policies.