Problem:

The goal of this problem is to empirically compare Prim's and Kruskal's Minimum Spanning Tree (MST) algorithms.

We will track both computer clock time and the number of operations, focusing specifically on key comparisons.

Input will be provided in a .txt file. See below for its structure. The first step of your code will be to read in this file and construct the Adjacency Table (preferred) or Cost Matrix for the undirected graph.

You may use any implementation of both algorithms, as well as pre-sort for Kruskal, but you should state in the beginning of your output which implementations you are using.

Your output should reflect the MST as it "grows" (after adding a vertex in Prim or adding an edge in Kruskal) as well as the content of the underlying tables so we can see the progress. Tracking the requisite measure of performance - both clock time and operations - as well as formatting your output in a clear and meaningful way is as much a part of your score as the working code itself.

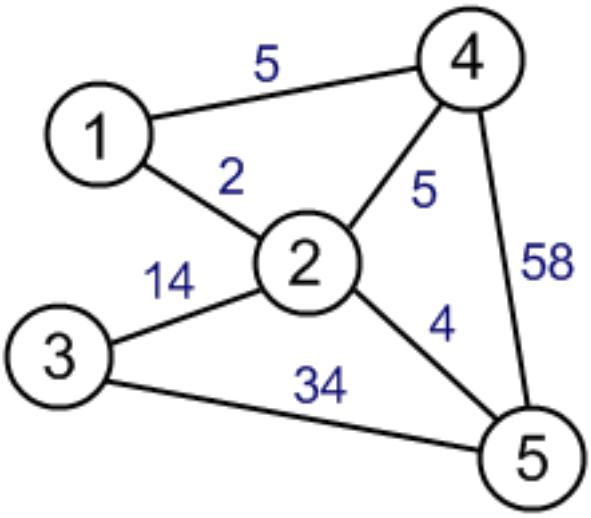
As a starting point, you may use any publicly available code - in the language of your choice - for the two algorithms:

https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/
https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

INPUT FORMAT IN NEXT PAGE

The Input

Input will be provided in a file input2.txt which resembles an Adjacency Table with costs. As an example, the input file that follows would correspond to this graph. In each line, first we have the vertex number, followed by the neighboring vertices and costs in an alternating fashion. The input given during peer-review will differ in content but adhere to the same structure (colon after each start vertex, commas separating end vertices, etc.), so your code should be prepared for that.



Sample input file

```
1 : 2 2, 4 5
2 : 1 2, 3 14, 4 5, 5 4
3 : 2 14, 5 34
4 : 1 5, 2 5, 5 58
5 : 2 4, 3 34, 4 58
```