Problem Statement : Longest MST edge. Run empirical studies to analyze the length of the longest edge in the MST and number of graph edges that are no longer than that one.

Solution : The above mentioned statement can be described by taking an example. Let's take a graph having Vertices "v" , Edges "e" which are connected to each other with their weights. N can be denoted as the number of edges that are lesser in number than the longest edge in the given Minimum Spanning Tree.

Procedure :

Step - 1 : Create a graph G.

Step - 2 : Denote the edge weights and insert them into the Binary Symbol table, bst.

Step - 3 : FInd the Minimum Spanning Tree and then insert edges of the MST in the Priority Queue in order to obtain the longest edge in the MST (maximum weighted edge) using the Kruskal's algorithm.

Step - 4 : We can find the rank from the Symbol table by using the getRank() method. The total count of edges less than the longest edge from the bst in represented by the rank.

Time Complexity : The time complexity to create a MST can be represented by E*logE with the help of Kruskal's algorithm.

Data Analysis :

Given Data is is represented by using graphs from test cases along with weights.

V = Vertices of the graph
E = Edges of the graph
N = number of edges lower then the longest edge in mst
Time Complexity $\propto$ ElogE
Proposed time complexity $\propto$ E+N*logV

Ratio (R) = ElogE/E+N*logV

V	E	N	ElogE	E+N*logV	R
8	16	12	64	52	1.23
250	1273	961	13130	8928.1	1.47
9	14	6	53.3	33	1.61
4	5	1	11.6	7	1.66
3	3	0	4.8	3	1.6

Conclusion :

- From the above mentioned analysis the ratio can be said as constant and is approximately equals to ≅ 1.6.
- The deviation increases according with the edges equal to longest edge in Minimum Spanning Tree.

References :

https://flylib.com/books/en/3.56.1.51/1/

https://en.wikipedia.org/wiki/Kruskal%27s_algorithm

The program can be accessed by visiting the Github link given below :

https://github.com/shubhamsahu02/ADS-2/tree/master/ADS-2-assignments/m24/MST