Mathematical foundations of Computer Science

Project Report

Submitted to:

Dr. Salman Abdul Moiz

Submitted by:

19MCMC55

19MCMC30

19MCMC16

18-11-2019

MCA - I

TABLE TO CONTENTS

INTRODUCTION

The project is based on finding a Minimum Spanning Tree using Kruskal’s Algorithm. The data set on which the MST was to be found is a list of the most frequently visited places inside the campus of the University of Hyderabad along with the distances (in Kilometers) that connected one place to another.

HYPOTHESIS

Since distance from a place to itself is always zero, therefore they were replaced by a constant value INFINITY (=999). This was done to prevent the inclusion of these zero values in the algorithm, since the algorithm operates on taking minimum cost (here, distance).

THEORY

Why is Kruskal’s algorithm needed?

➔ Kruskal’s algorithm is needed as it constructs a MST which finds the minimum distance required to travel to all the frequently visited places. It uses a greedy technique (i.e., it chooses the one which is the most optimal at the moment only) since it sorts the distances in non decreasing order and selects the one with minimum distance.

  Running time of the Algorithm: O(E logV)

How does Kruskal’s algorithm work?

1. List all the edges of the graph in increasing order of weight.
2. Select the smallest edge of graph G, this is the first edge of spanning tree T.
3. If there are more than one edge of the same minimal value, select one of them arbitrarily.
4. Select another edge of minimal value from the remaining edges such that inclusion of this edge does not form a circuit.
5. Repeat step 3 until the the tree T contains N-1 edges.

DATA

We have arbitrarily taken 38 most frequently places in HCU campus and are listed below:

0	Main Gate	1	Administrative Building
2	State Bank of India	3	Health Centre
4	North Shopping Complex	5	Zakhir Hussain LHC
6	S. Radhakrishnan LHC	7	Student Centre
8	SCIS	9	DST Auditorium
10	IGM Library	11	School of management studies
12	School of social sciences	13	Ambedkar LHC
14	Electronics Building	15	School of humanities
16	Womens Hostel	17	GOPS
18	Sarojini Naidu school of arts and com	19	School of Chemistry
20	CR rao institute	21	CMSD
22	Football Ground	23	Tennis Court
24	Yoga Centre	25	Mens Hostel A-D
26	Gymnasium	27	Shooting Range
28	School of Medical science	29	Post Office
30	Chief Warden office	31	Automobile Workshop
32	Tagore International Hostel	33	South Campus Shopping Complex
34	Centre for integrated studies	35	School of life sciences
36	Mushroom Rock	37	Teacup junction

We then constructed a 38X38 matrix which a representation of undirected graph which depicts distance from one place to another respectively.

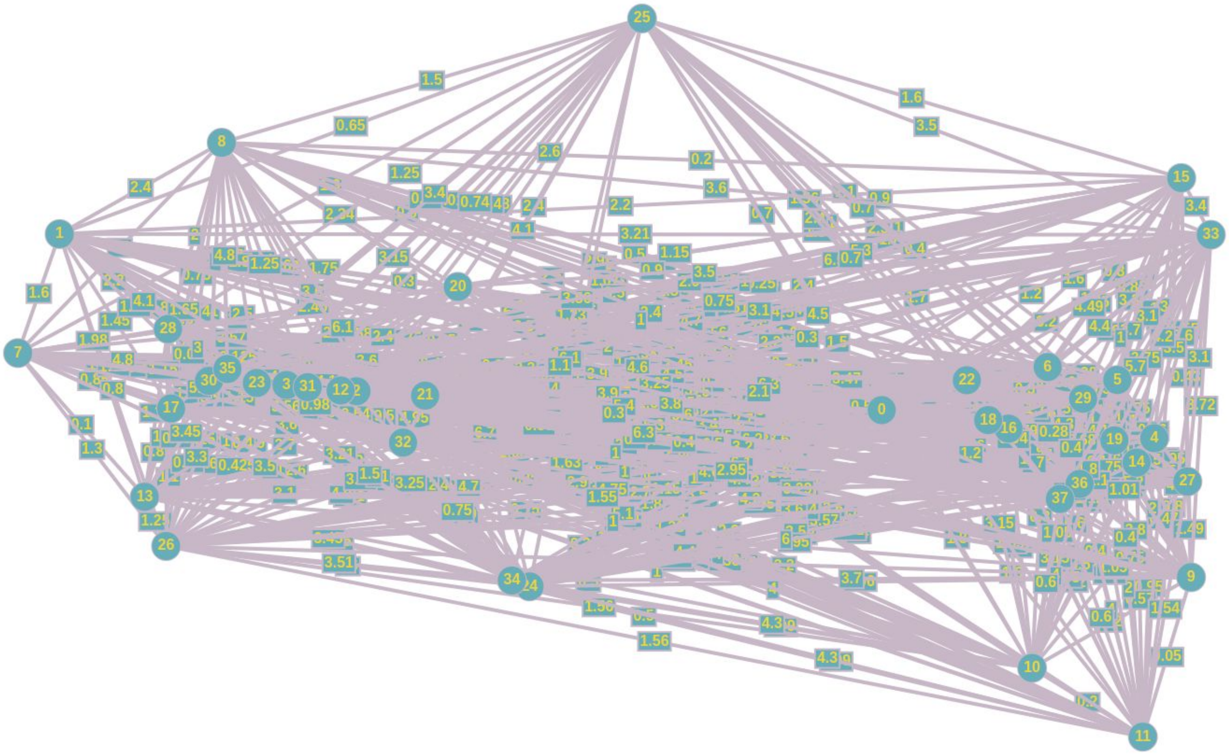
PSEUDOCODE

MST-KRUSKAL (G):

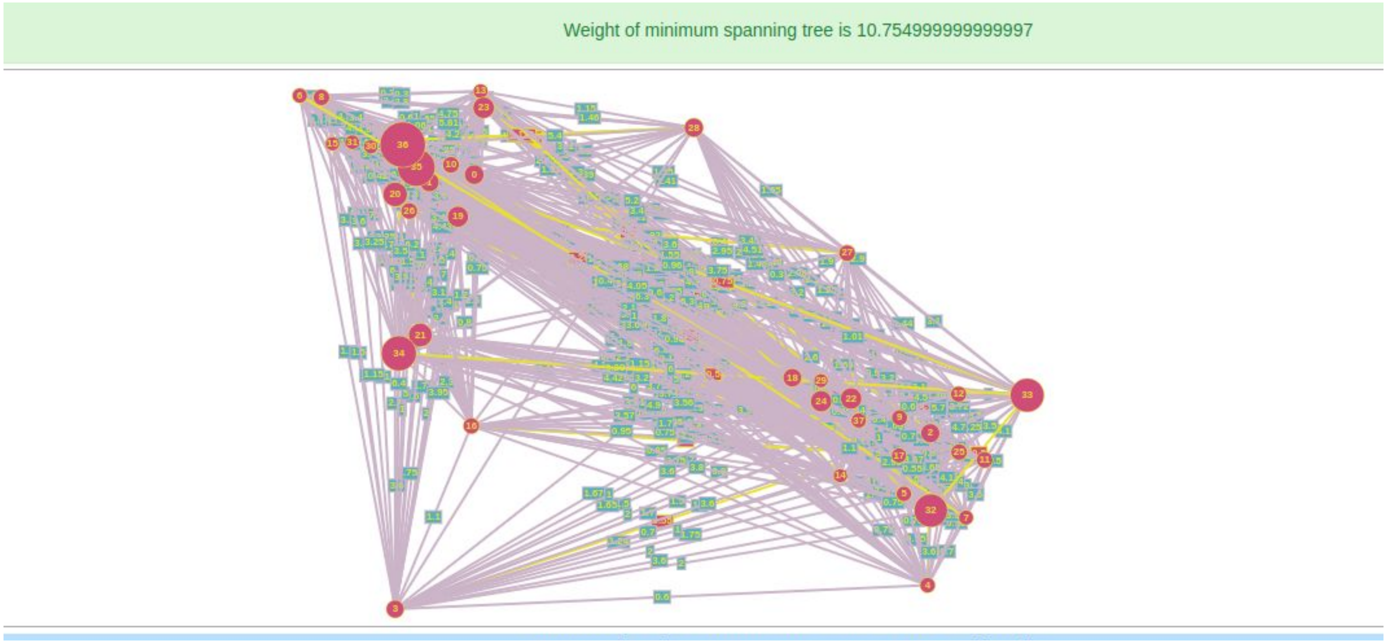
```
● G1=0
● for every vertex v in G.v do:
    ○ Make-set (v)                //makes a new set whose sole member is v
● for every edge (u,v) in G.e ordered by increasing weight(u,v):
    ○ if Find(u) != Find(v):
    ○ G1 = G1 U {(u,v)}
    ○ Union(u,v)
● return G1
```

RESULTS

The graph formed for the matrix, that includes edges for every edge set (u,v) present is as follows:



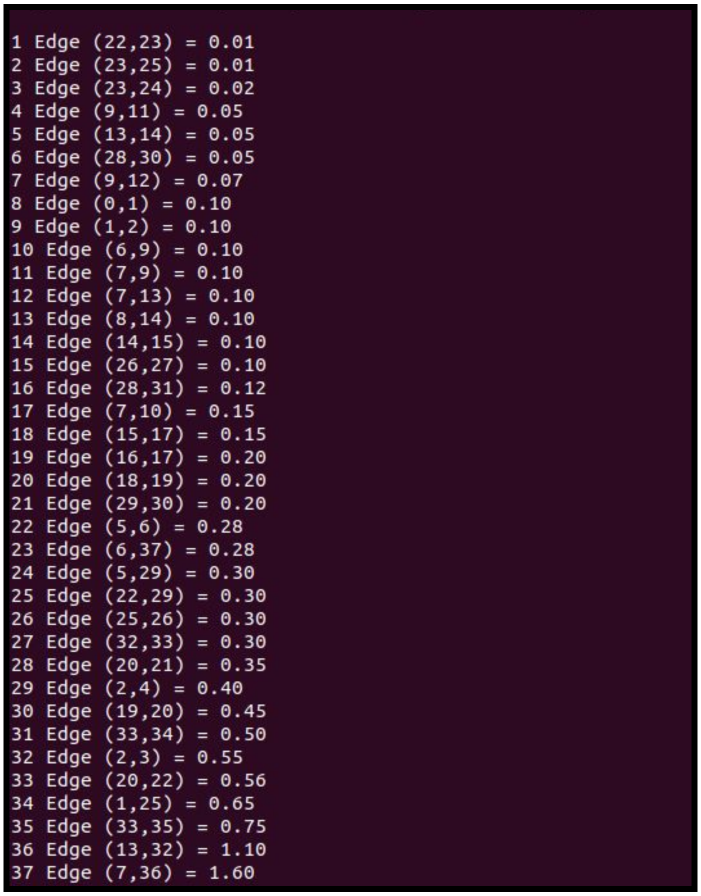
Due to a large number of nodes its difficult to determine all the edges therefore we have highlighted the minimum spanning tree in the next image:



Weight of minimum spanning tree is 10.754999999999997

The Minimum Spanning Tree is highlighted with the nodes in red and the minimal edges in yellow color.

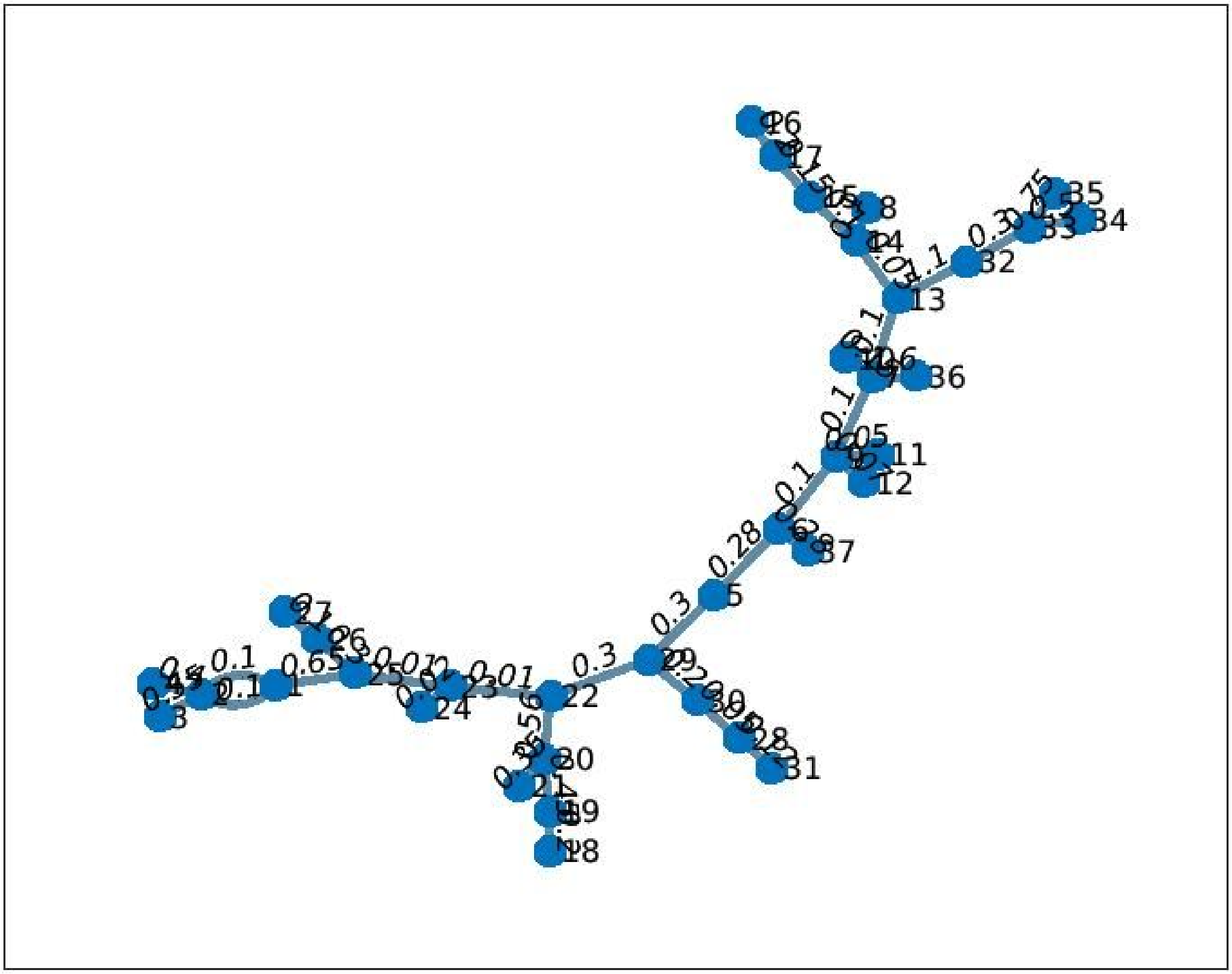
The edges that were selected in non decreasing order are as follows:



```
1 Edge (22,23) = 0.01
2 Edge (23,25) = 0.01
3 Edge (23,24) = 0.02
4 Edge (9,11) = 0.05
5 Edge (13,14) = 0.05
6 Edge (28,30) = 0.05
7 Edge (9,12) = 0.07
8 Edge (0,1) = 0.10
9 Edge (1,2) = 0.10
10 Edge (6,9) = 0.10
11 Edge (7,9) = 0.10
12 Edge (7,13) = 0.10
13 Edge (8,14) = 0.10
14 Edge (14,15) = 0.10
15 Edge (26,27) = 0.10
16 Edge (28,31) = 0.12
17 Edge (7,10) = 0.15
18 Edge (15,17) = 0.15
19 Edge (16,17) = 0.20
20 Edge (18,19) = 0.20
21 Edge (29,30) = 0.20
22 Edge (5,6) = 0.28
23 Edge (6,37) = 0.28
24 Edge (5,29) = 0.30
25 Edge (22,29) = 0.30
26 Edge (25,26) = 0.30
27 Edge (32,33) = 0.30
28 Edge (20,21) = 0.35
29 Edge (2,4) = 0.40
30 Edge (19,20) = 0.45
31 Edge (33,34) = 0.50
32 Edge (2,3) = 0.55
33 Edge (20,22) = 0.56
34 Edge (1,25) = 0.65
35 Edge (33,35) = 0.75
36 Edge (13,32) = 1.10
37 Edge (7,36) = 1.60
```

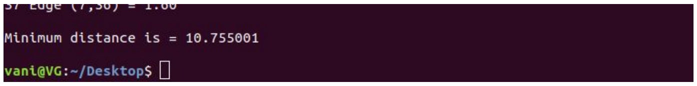
As it is visible, they're selected according to weights even if they're disjointed.

The final MST on removing all the extra edges is given below:



CONCLUSION

The conclusion drawn from the extensive data that was collected from the website as well as manually is that the minimum distance (or cost) is roughly equal to 10.75 ~ 11 Kilometers.



```
37 Edge (7,36) = 1.60

Minimum distance is = 10.755001

vani@VG:~/Desktop$ 
```

OS USED

Ubuntu 18.04.3

SOFTWARES AND TOOLS USED

1. Online compiler - https://www.codechef.com/ide
2. MATLAB - https://in.mathworks.com/help/matlab/ref/graph.plot.html
3. Graph online - https://graphonline.ru/en/
4. CS Academy - https://csacademy.com/app/graph_editor/

REFERENCES

1. Introduction to Algorithms, 3rd Ed., Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein
2. Class notes
3. Visual Algo - https://visualgo.net/en/mst
4. Youtube - https://www.youtube.com/watch?v=Yo7sddEVONg
5. Wiki - https://en.wikipedia.org/wiki/Kruskal%27s_algorithm