Spanning Tree

Amith C A

2020MCS120003

DSC522 Project 2

Minimum Spanning tree

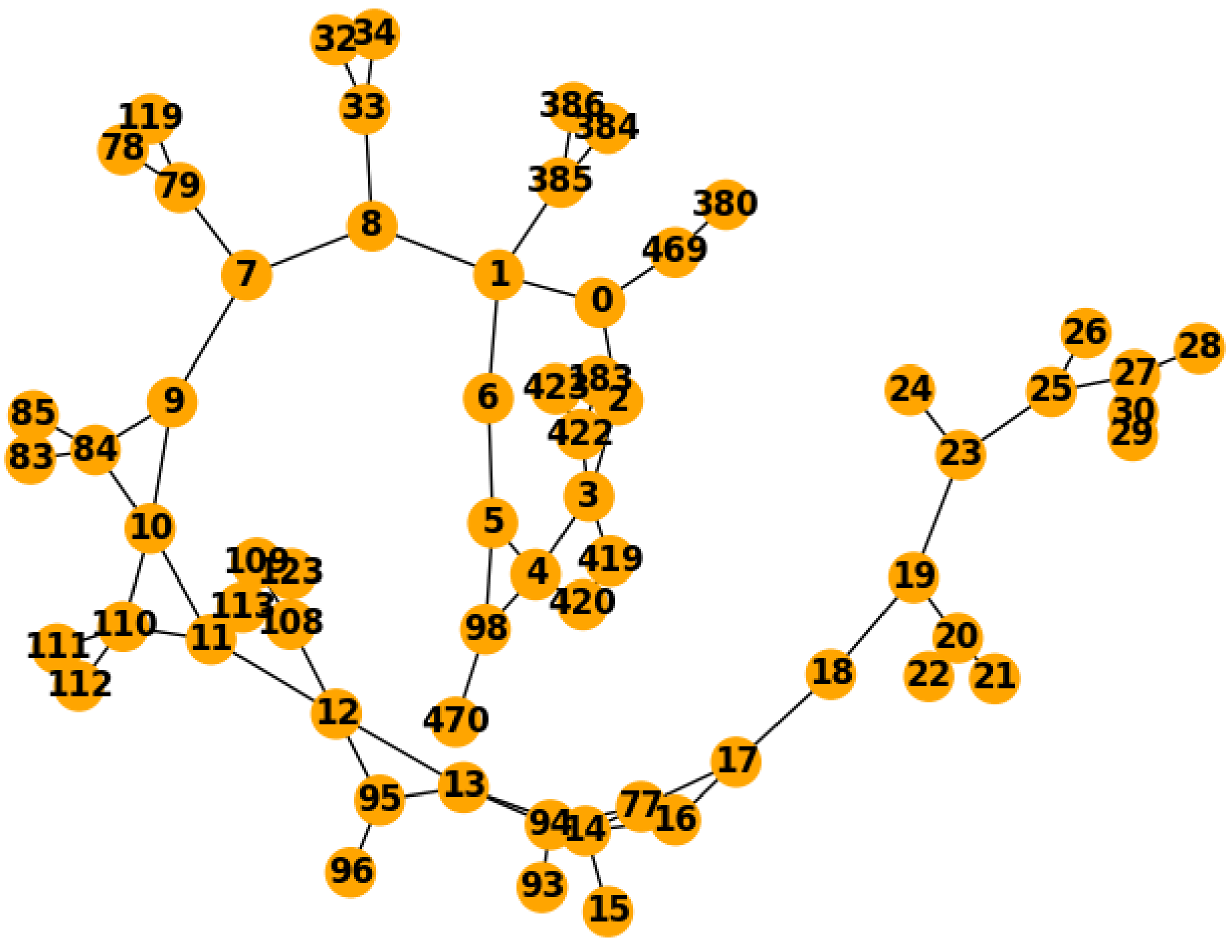
A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible. More generally, any edge-weighted undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of the minimum spanning trees for its connected components.

Input Graph

Consider the data roadNet-CA.txt, which shows the road network data from California.

The graph constructed using the data from roadNet-CA.txt is shown below, which serves as the input graph to our algorithms.

Input graph plot:



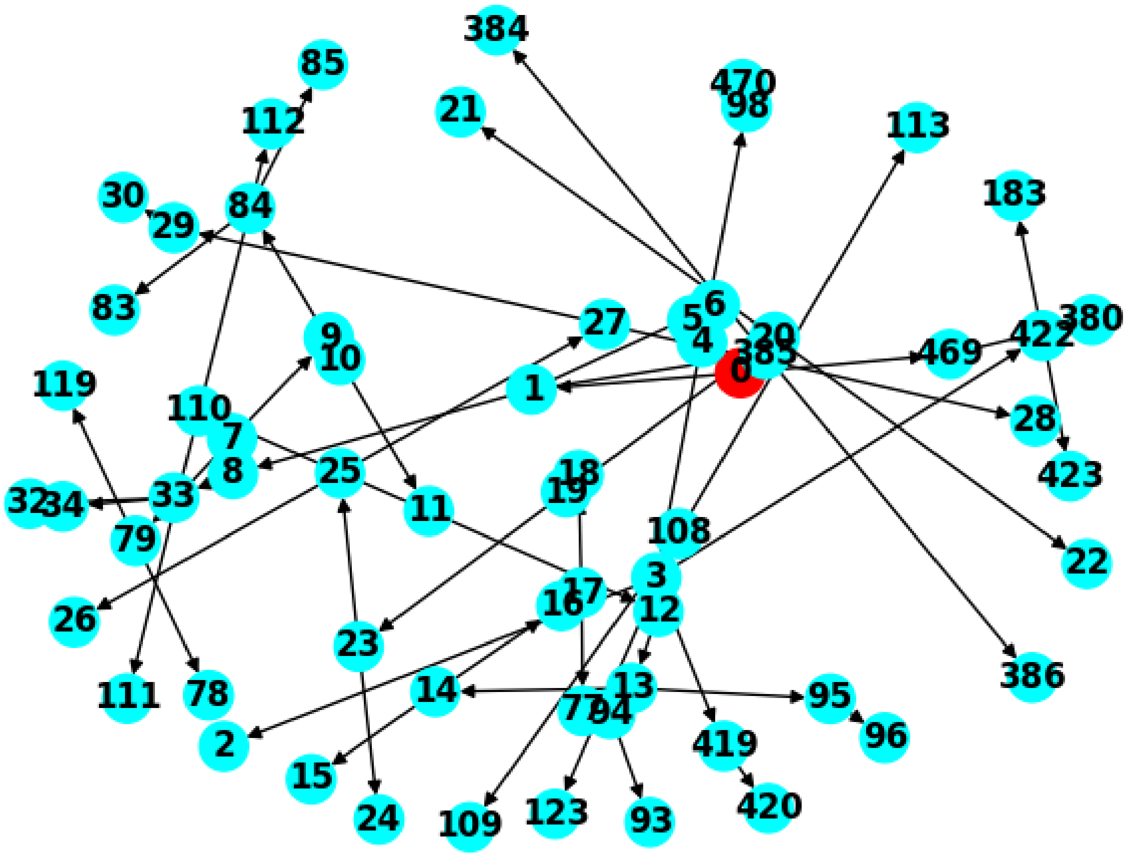
Depth First Search

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.Depth-first search is like walking through a corn maze. You explore one path, hit a dead end, and go back and try a different one

Uses of DFS :
Find path between two given vertices u and v.
Perform topological sorting is used to scheduling jobs from given dependencies among jobs.
Find strongly connected components of a graph.

The DFS algorithm is a recursive algorithm that uses the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, else by backtracking.
Here, the word backtrack means that when you are moving forward and there are no more nodes along the current path, you move backwards on the same path to find nodes to traverse. All the nodes will be visited on the current path till all the unvisited nodes have been traversed after which the next path will be selected.
This recursive nature of DFS can be implemented using stacks. The basic idea is as follows:
Pick a starting node and push all its adjacent nodes into a stack.
Pop a node from stack to select the next node to visit and push all its adjacent nodes into a stack.
Repeat this process until the stack is empty. However, ensure that the nodes that are visited are marked. This will prevent you from visiting the same node more than once. If you do not mark the nodes that are visited and you visit the same node more than once, you may end up in an infinite loop.

DFS Tree



384
85
112
21
470
98
113
183
30
29
84
83
27
5
6
4
20
385
0
469
422
380
9
10
1
119
110
7
8
25
28
423
32
34
33
79
18
19
11
108
22
26
17
16
3
12
23
14
13
95
96
386
111
78
2
77
94
419
15
24
109
123
93
420

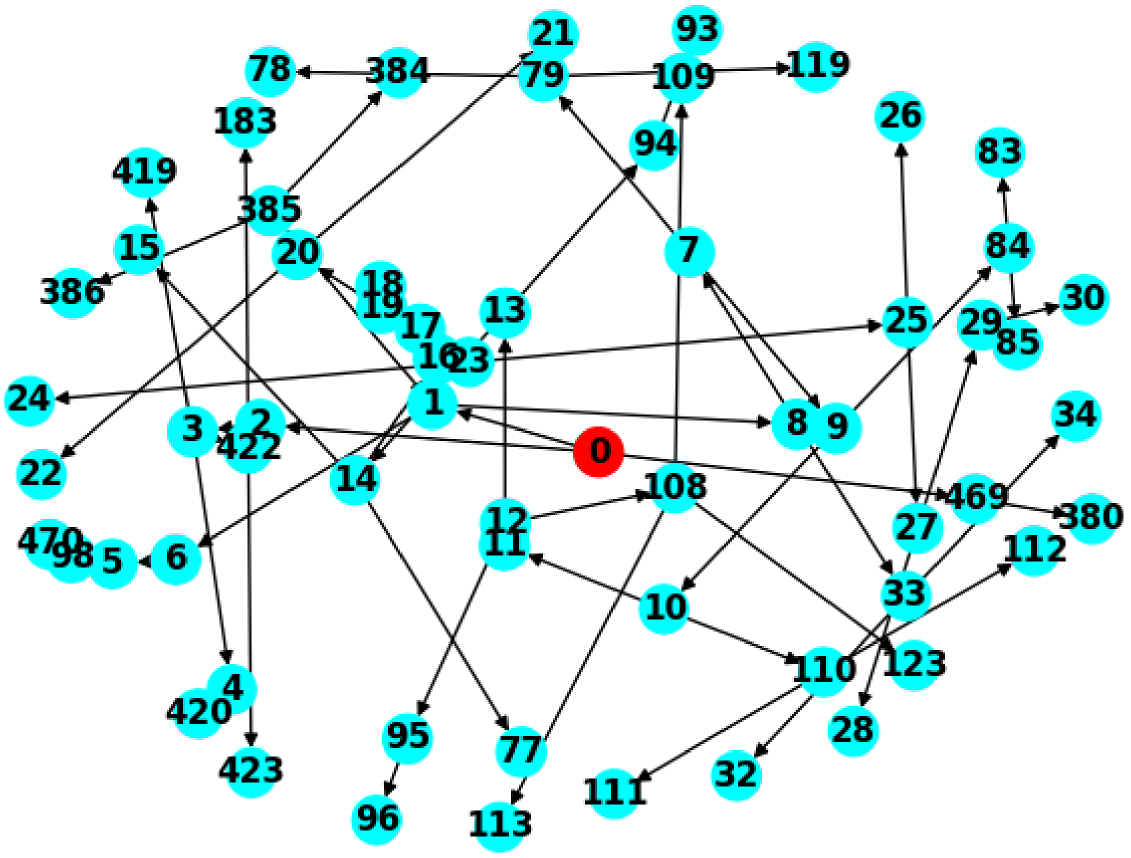
Breadth First Search

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

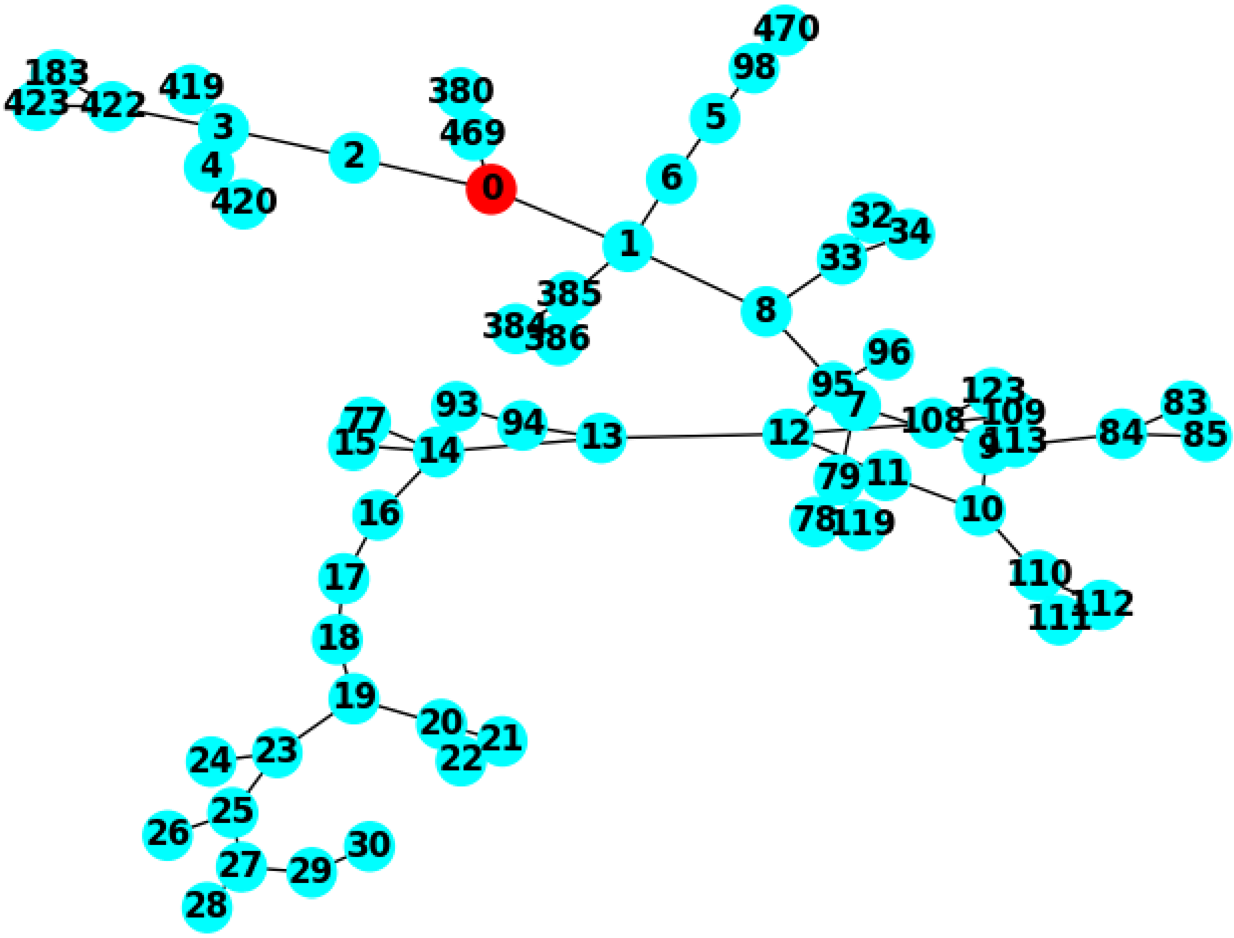
It uses the opposite strategy of depth-first search, which instead explores the node branch as far as possible before being forced to backtrack and expand other nodes.

BFS and its application in finding connected components of graphs were invented in 1945 by Konrad Zuse, in his (rejected) Ph.D. thesis on the PlankalkÃ¼l programming language, but this was not published until 1972. It was reinvented in 1959 by Edward F. Moore, who used it to find the shortest path out of a maze, and later developed by C. Y. Lee into a wire routing algorithm (published 1961).

BFS Tree


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
32
33
34
77
78
79
83
84
85
93
94
95
96
98
108
109
110
111
112
113
119
123
183
380
384
385
386
419
420
422
423
469
470

Minimum spanning Tree



Conclusion

Successfully implemented the minimum spanning tree