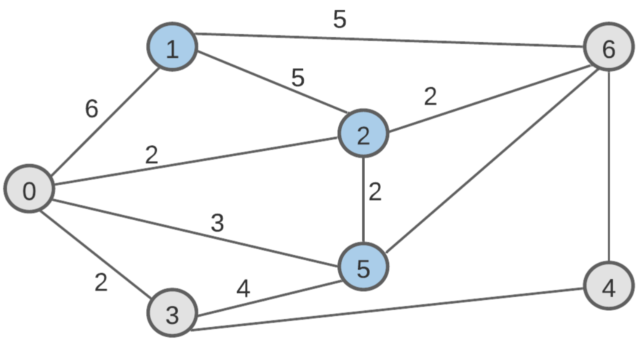
Implementation of Steiner Tree

This document contains implementation of Steiner Tree using Dynamic Programming approach on various inputs.

Case 1. For this case we will walk through all the steps one by one so as to have the understanding of code.

Consider the following graph* as an input.



Terminal Vertices = { 0, 3, 4, 6}

K = 4

First we will check for assumptions:

Assumption 1: K = 4 which is greater than 1.

Assumption 2: Graph is connected; for this we can either use any graph traversal(DFS/BFS) or since we will be using Floyd Warshall to calculate shortest distance path between all vertices we can also use that.

Assumption 3: Terminal vertices should have degree exactly1 and no two terminal vertices should be adjacent, to achieve this property, for every terminal t ∈ T, we attach a new neighbour t′of degree 1, that is, we create a new vertex t′and an edge tt′of some fixed weight, say 1. Also our modified graph and original graph are isomorphic.

This assumption is valid because all the leaves in the Steiner tree must be terminals. Otherwise, one could simply delete the non-terminal leaves, yielding a feasible solution with less cost.

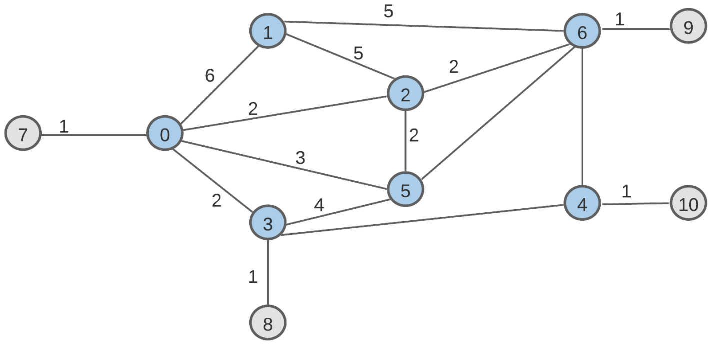
* Code doesn't visualises this graph, it is built manually using https://lucid.app/lucidspark/invitations/accept/inv_6e7dff3f-43ee-4c3a-8f36-3eb1c97356bf

So as to satisfy assumption 3 we will add 4 extra vertices {7, 8, 9, 10}

And edges {(0,7), (3, 5), (5, 6), (6, 4)}

Our intermediate graph* looks like:

```
Enter number of vertices and edges in the graph : 7 12
Enter vertices forming edges and their respective weight
0 1 6
0 2 2
0 5 3
0 3 2
1 2 5
1 6 5
2 6 2
2 5 2
3 5 4
3 4 13
4 6 4
5 6 3
Enter number of Terminal verlices : 4
Enter Terminal Vertices : 0 3 4 6

Our Graph is created!!!
----------------------------Adjacency Matrix---------------------------
     0     6     2     2   INF     3   INF
     6     0     5   INF   INF   INF     5
     2     5     0   INF   INF     2     2
     2   INF   INF     0    13     4   INF
   INF   INF   INF    13     0   INF     4
     3   INF     2     4   INF     0     3
   INF     5     2   INF     4     3     0

Now we will check for our 3 assupmtions

Assumption 1 passed!!!
 i.e k > 1

Assumption 2 passed!!!
 i.e our graph is connected
     0     6     2     2   INF     3   INF     1   INF   INF   INF
     6     0     5   INF   INF   INF     5   INF   INF   INF   INF
     2     5     0   INF   INF     2     2   INF   INF   INF   INF
     2   INF   INF     0    13     4   INF   INF     1   INF   INF
   INF   INF   INF    13     0   INF     4   INF   INF     1   INF
     3   INF     2     4   INF     0     3   INF   INF   INF   INF
   INF     5     2   INF     4     3     0   INF   INF   INF     1
     1   INF   INF   INF   INF   INF   INF     0   INF   INF   INF
   INF   INF   INF     1   INF   INF   INF   INF     0   INF   INF
   INF   INF   INF   INF     1   INF   INF   INF   INF     0   INF
   INF   INF   INF   INF   INF   INF     1   INF   INF   INF     0

Now, all the terminal nodes have degree one and are not adjacent
```

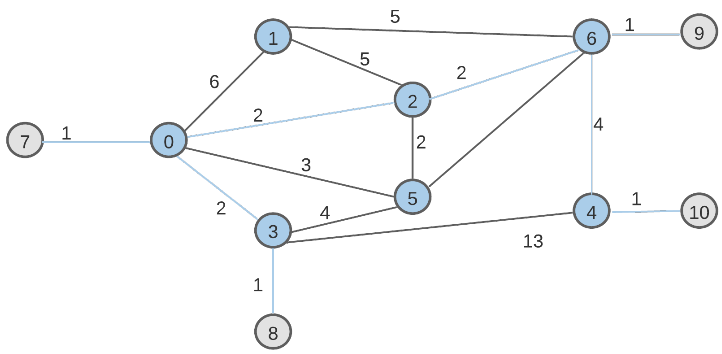


Now when our graph satisfy all the assumptions, still an important pre processing step is left i.e. we need to calculate the shortest path between vertices, for this purpose I have used Floyd Warshall because it was flexible to use.

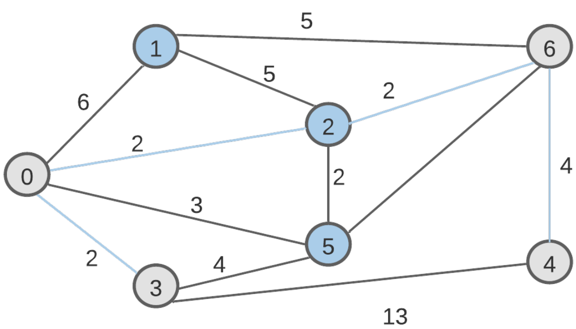
```
--------------------All Pair Shortest Path Matrix--------------------
    0     6     2     2     8     3     4     1     3     9     5
    6     0     5     8     9     7     5     7     9    10     6
    2     5     0     4     6     2     2     3     5     7     3
    2     8     4     0    10     4     6     3     1    11     7
    8     9     6    10     0     7     4     9    11     1     5
    3     7     2     4     7     0     3     4     5     8     4
    4     5     2     6     4     3     0     5     7     5     1
    1     7     3     3     9     4     5     0     4    10     6
    3     9     5     1    11     5     7     4     0    12     8
    9    10     7    11     1     8     5    10    12     0     6
    5     6     3     7     5     4     1     6     8     6     0
```

*In ppt I didn't elaborate this step, as the example I chose was already pre processed, because main motive was the understanding of algorithm.

The last step is to apply DP algorithm, table will have 2^k rows = 16 for terminal vertices as {7, 8, 9, 10} and column represents vertices V(G)\ T = {0, 1, 2, 3, 4, 5, 6}



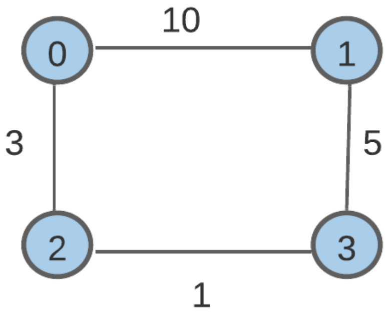
We will obtain this as a Steiner Tree(indicated by blue Lines) but since we added k extra edges each of weight 1 we will not consider that as a part of our output.



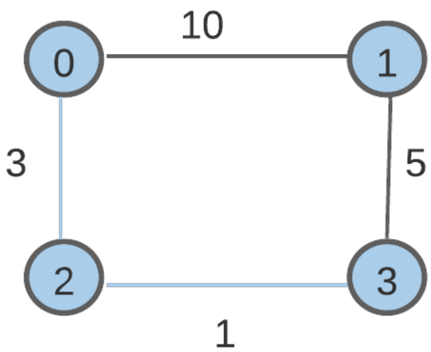
This will be our final answer.

```
-------------------------------DP Table-------------------------------
   INF   INF   INF   INF   INF   INF   INF
     5     6     3     7     5     4     1
     9    10     7    11     1     8     5
    10    11     8    12     6     9     6
     3     9     5     1    11     5     7
     8    13     8     8    12     9     8
    12    17    12    12    12    13    12
    13    18    13    13    13    14    13
     1     7     3     3     9     4     5
     6    11     6     8    10     8     6
    10    15    10    12    10    12    10
    11    16    11    13    11    13    11
     4    10     6     4    12     7     8
     9    14     9     9    13    11     9
    13    18    13    13    13    15    13
    14    19    14    14    14    16    14
Cost of Minimum Steiner Tree = 10
```

Case 2: Graph where |V| = |E| = 4

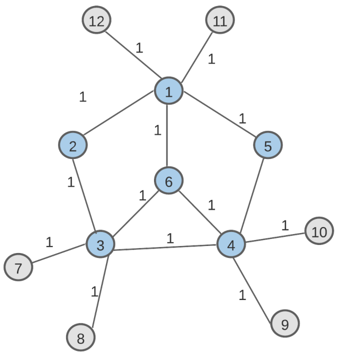



```
Enter number of vertices and edges in the graph : 4 4
Enter vertices forming edges and their respective weight
0 1 10
0 2 3
2 3 1
3 1 5
Enter number of Terminal verlices : 2
Enter Terminal Vertices : 0 3

Our Graph is created!!!
----------------------------Adjacency Matrix-------------------------
     0    10     3   INF
    10     0   INF     5
     3   INF     0     1
   INF     5     1     0

Now we will check for our 3 assupmtions

Assumption 1 passed!!!
 i.e k > 1

Assumption 2 passed!!!
 i.e our graph is connected
     0    10     3   INF     1   INF
    10     0   INF     5   INF   INF
     3   INF     0     1   INF   INF
   INF     5     1     0   INF     1
     1   INF   INF   INF     0   INF
   INF   INF   INF     1   INF     0

Now, all the terminal nodes have degree one and are not adjacent
```




```
---------------------All Pair Shortest Path Matrix---------------------
     0     9     3     4     1     5
     9     0     6     5    10     6
     3     6     0     1     4     2
     4     5     1     0     5     1
     1    10     4     5     0     6
     5     6     2     1     6     0

------------------------------DP Table------------------------------
   INF   INF   INF   INF
     5     6     2     1
     1    10     4     5
     6    11     6     6
Cost of Minimum Steiner Tree = 4
Program ended with exit code: 0
```

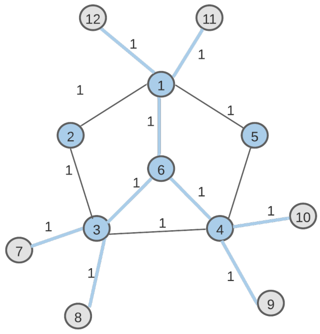
Case 3: Graph where $|V| = 13$, $|E| = 14$



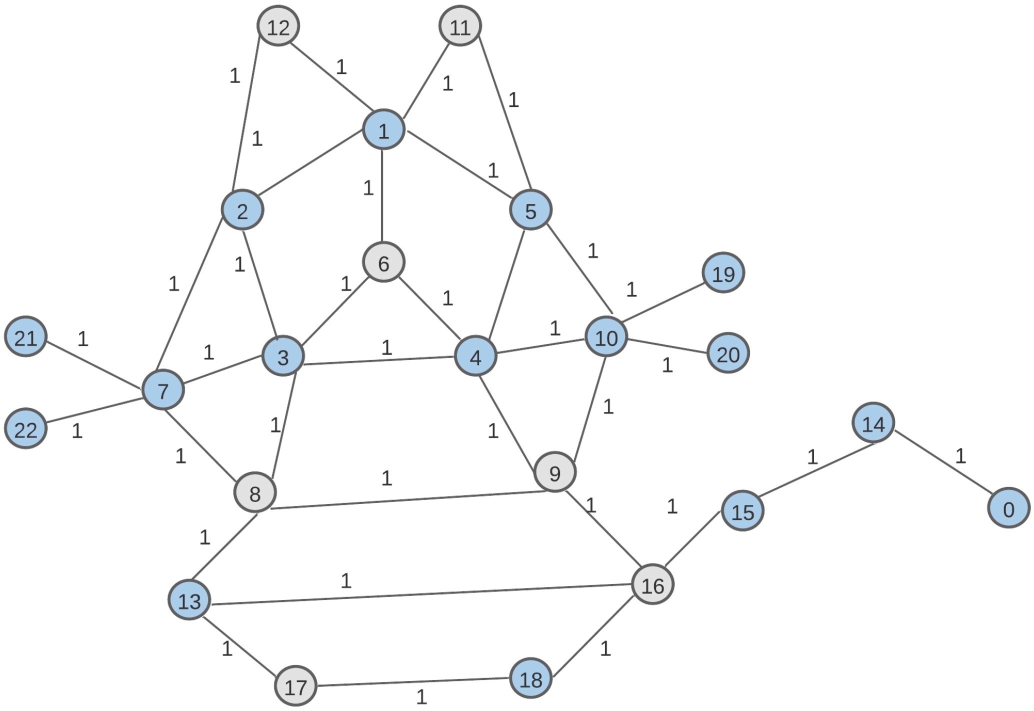

```
Enter number of vertices and edges in the graph : 12 14
Enter vertices forming edges and their respective weight
0 5 1
0 4 1
0 1 1
1 2 1
2 5 1
2 3 1
2 6 1
2 7 1
3 5 1
3 8 1
3 9 1
3 4 1
0 11 1
0 10 1
Enter number of Terminal verlices : 6
Enter Terminal Vertices : 6 7 8 9 10 11

Our Graph is created!!!
----------------------------Adjacency Matrix---------------------------
     0     1   INF   INF     1     1   INF   INF   INF   INF     1     1
     1     0     1   INF   INF   INF   INF   INF   INF   INF   INF   INF
   INF     1     0     1   INF     1     1     1   INF   INF   INF   INF
   INF   INF     1     0     1     1   INF   INF     1     1   INF   INF
     1   INF   INF     1     0   INF   INF   INF   INF   INF   INF   INF
     1   INF     1     1   INF     0   INF   INF   INF   INF   INF   INF
   INF   INF     1   INF   INF   INF     0   INF   INF   INF   INF   INF
   INF   INF     1   INF   INF   INF   INF     0   INF   INF   INF   INF
   INF   INF   INF     1   INF   INF   INF   INF     0   INF   INF   INF
   INF   INF   INF     1   INF   INF   INF   INF   INF     0   INF   INF
     1   INF   INF   INF   INF   INF   INF   INF   INF   INF     0   INF
     1   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     0
```

```
Now we will check for our 3 assupmtions

Assumption 1 passed!!!
 i.e k > 1

Assumption 2 passed!!!
 i.e our graph is connected
     0     1   INF   INF     1     1   INF   INF   INF   INF     1     1   INF
         INF   INF   INF   INF   INF
     1     0     1   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
         INF   INF   INF   INF   INF
   INF     1     0     1   INF     1     1     1   INF   INF   INF   INF   INF
       INF   INF   INF   INF   INF
   INF   INF     1     0     1     1   INF   INF     1     1   INF   INF   INF
       INF   INF   INF   INF   INF
     1   INF   INF     1     0   INF   INF   INF   INF   INF   INF   INF   INF
         INF   INF   INF   INF   INF
     1   INF     1     1   INF     0   INF   INF   INF   INF   INF   INF   INF
         INF   INF   INF   INF   INF
   INF   INF     1   INF   INF   INF     0   INF   INF   INF   INF   INF     1
       INF   INF   INF   INF   INF
   INF   INF     1   INF   INF   INF   INF     0   INF   INF   INF   INF   INF
         1   INF   INF   INF   INF
   INF   INF   INF     1   INF   INF   INF   INF     0   INF   INF   INF   INF
       INF     1   INF   INF   INF
   INF   INF   INF     1   INF   INF   INF   INF   INF     0   INF   INF   INF
       INF   INF     1   INF   INF
     1   INF   INF   INF   INF   INF   INF   INF   INF   INF     0   INF   INF
         INF   INF   INF     1   INF
     1   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     0   INF
         INF   INF   INF   INF     1
   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF   INF     0
       INF   INF   INF   INF   INF
   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF   INF
         0   INF   INF   INF   INF
   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF   INF
         0   INF   INF   INF   INF
   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF
       INF     0   INF   INF   INF
   INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF
       INF   INF     0   INF   INF
   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF
       INF   INF   INF     0   INF
   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF
       INF   INF   INF   INF     0

Now, all the terminal nodes have degree one and are not adjacent
```

```
------------------------------DP Table------------------------------
   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
     2     3     4     4     3     3     5     5     5     5     3     1
     2     3     4     4     3     3     5     5     5     5     1     3
     4     5     6     6     5     5     7     7     7     7     4     4
     4     4     3     2     3     3     4     4     3     1     5     5
     6     7     7     6     6     6     8     8     7     6     7     6
     6     7     7     6     6     6     8     8     7     6     6     7
     8     9     9     8     8     8    10    10     9     8     8     8
     4     4     3     2     3     3     4     4     1     3     5     5
     6     7     7     6     6     6     8     8     6     7     7     6
     6     7     7     6     6     6     8     8     6     7     6     7
     8     9     9     8     8     8    10    10     8     9     8     8
     6     6     5     4     5     5     6     6     4     4     7     7
     8     9     9     8     8     8    10    10     8     8     9     8
     8     9     9     8     8     8    10    10     8     8     8     9
    10    11    11    10    10    10    12    12    10    10    10    10
     4     3     2     3     4     3     3     1     4     4     5     5
     6     6     6     7     7     6     7     6     8     8     7     6
     6     6     6     7     7     6     7     6     8     8     6     7
     8     8     8     9     9     8     9     8    10    10     8     8
     7     6     5     5     6     6     6     5     6     5     8     8
     9     9     9     9     9     9    10     9    10     9    10     9
     9     9     9     9     9     9    10     9    10     9     9    10
    11    11    11    11    11    11    12    11    12    11    11    11
     7     6     5     5     6     6     6     5     5     6     8     8
     9     9     9     9     9     9    10     9     9    10    10     9
     9     9     9     9     9     9    10     9     9    10     9    10
    11    11    11    11    11    11    12    11    11    12    11    11
     9     8     7     7     8     8     8     7     7     7    10    10
    11    11    11    11    11    11    12    11    11    11    12    11
    11    11    11    11    11    11    12    11    11    11    11    12
    13    13    13    13    13    13    14    13    13    13    13    13
     4     3     2     3     4     3     1     3     4     4     5     5
     6     6     6     7     7     6     6     7     8     8     7     6
     6     6     6     7     7     6     6     7     8     8     6     7
     8     8     8     9     9     8     8     9    10    10     8     8
     7     6     5     5     6     6     5     6     6     5     8     8
     9     9     9     9     9     9     9    10    10     9    10     9
     9     9     9     9     9     9     9    10    10     9     9    10
    11    11    11    11    11    11    11    12    12    11    11    11
     7     6     5     5     6     6     5     6     5     6     8     8
     9     9     9     9     9     9     9    10     9    10    10     9
     9     9     9     9     9     9     9    10     9    10     9    10
    11    11    11    11    11    11    11    12    11    12    11    11
     9     8     7     7     8     8     7     8     7     7    10    10
    11    11    11    11    11    11    11    12    11    11    12    11
    11    11    11    11    11    11    11    12    11    11    11    12
    13    13    13    13    13    13    13    14    13    13    13    13
     6     5     4     5     6     5     4     4     6     6     7     7
     8     8     8     9     9     8     8     8    10    10     9     8
     8     8     8     9     9     8     8     8    10    10     8     9
    10    10    10    11    11    10    10    10    12    12    10    10
     9     8     7     7     8     8     7     7     8     7    10    10
    11    11    11    11    11    11    11    11    12    11    12    11
    11    11    11    11    11    11    11    11    12    11    11    12
    13    13    13    13    13    13    13    13    14    13    13    13
     9     8     7     7     8     8     7     7     7     8    10    10
    11    11    11    11    11    11    11    11    11    12    12    11
    11    11    11    11    11    11    11    11    11    12    11    12
    13    13    13    13    13    13    13    13    13    14    13    13
    11    10     9     9    10    10     9     9     9     9    12    12
    13    13    13    13    13    13    13    13    13    13    14    13
    13    13    13    13    13    13    13    13    13    13    13    14
    15    15    15    15    15    15    15    15    15    15    15    15
Cost of Minimum Steiner Tree = 9
Program ended with exit code: 0
```



Case 4: Graph , |V| = 23, |E| = 34




```
Enter number of vertices and edges in the graph : 23 34
Enter vertices forming edges and their respective weight
12 1 1
11 1 1
12 2 1
11 5 1
1 5 1
1 2 1
1 6 1
2 7 1
2 3 1
3 6 1
6 4 1
5 4 1
5 10 1
21 7 1
22 7 1
7 3 1
3 4 1
4 10 1
10 19 1
10 20 1
7 8 1
3 8 1
8 9 1
4 9 1
9 10 1
8 13 1
9 16 1
13 16 1
16 15 1
15 14 1
14 0 1
13 17 1
17 18 1
18 16 1
```

```
Enter number of Terminal verlices : 7
Enter Terminal Vertices : 6 8 9 11 12 16 17

Our Graph is created!!!
----------------------------Adjacency Matrix---------------------------
    0   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
        INF     1   INF   INF   INF   INF   INF   INF   INF   INF
  INF     0     1   INF   INF     1     1   INF   INF   INF   INF     1     1
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF     1     0     1   INF   INF   INF     1   INF   INF   INF   INF     1
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF     1     0     1   INF     1     1     1   INF   INF   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF   INF     1     0     1     1   INF   INF     1     1   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF     1   INF   INF     1     0   INF   INF   INF   INF     1     1   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF     1   INF     1     1   INF     0   INF   INF   INF   INF   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF     1     1   INF   INF   INF     0     1   INF   INF   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF     1     1
  INF   INF   INF     1   INF   INF   INF     1     0     1   INF   INF   INF
        1   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF     1   INF   INF   INF     1     0     1   INF   INF
      INF   INF   INF     1   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF     1     1   INF   INF   INF     1     0   INF   INF
      INF   INF   INF   INF   INF   INF     1     1   INF   INF
  INF     1   INF   INF   INF     1   INF   INF   INF   INF   INF     0   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF     1     1   INF   INF   INF   INF   INF   INF   INF   INF   INF     0
      INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF
        0   INF   INF     1     1   INF   INF   INF   INF   INF
    1   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
        INF     0     1   INF   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
      INF     1     0     1   INF   INF   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF
        1   INF     1     0   INF     1   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
        1   INF   INF   INF     0     1   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF   INF
      INF   INF   INF     1     1     0   INF   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF
      INF   INF   INF   INF   INF   INF     0   INF   INF   INF
  INF   INF   INF   INF   INF   INF   INF   INF   INF   INF     1   INF   INF
      INF   INF   INF   INF   INF   INF   INF     0   INF   INF
  INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF     0   INF
  INF   INF   INF   INF   INF   INF   INF     1   INF   INF   INF   INF   INF
      INF   INF   INF   INF   INF   INF   INF   INF   INF     0
```

Now we will check for our 3 assupmtions

Assumption 1 passed!!!
i.e k > 1

Assumption 2 passed!!!
i.e our graph is connected

0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	0	1	INF	INF	1	1	INF	INF	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	1	0	1	INF	INF	INF	1	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	1	0	1	INF	1	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	1	0	1	1	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	1	INF	INF	1	0	INF	INF	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	1	INF	1	1	INF	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF
INF	INF	1	1	INF	INF	INF	0	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	1	INF	INF	INF	1	0	1	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF
INF	INF	INF	INF	1	INF	INF	INF	1	0	1	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF
INF	INF	INF	INF	1	1	INF	INF	INF	1	0	INF	INF	INF	INF	INF	INF	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	1	INF	INF	INF	1	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF
INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	0	INF	INF	1	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	0	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	1	INF	1	0	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	0	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	1	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0	INF
INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	1	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	0

Now, all the terminal nodes have degree one and are not adjacent

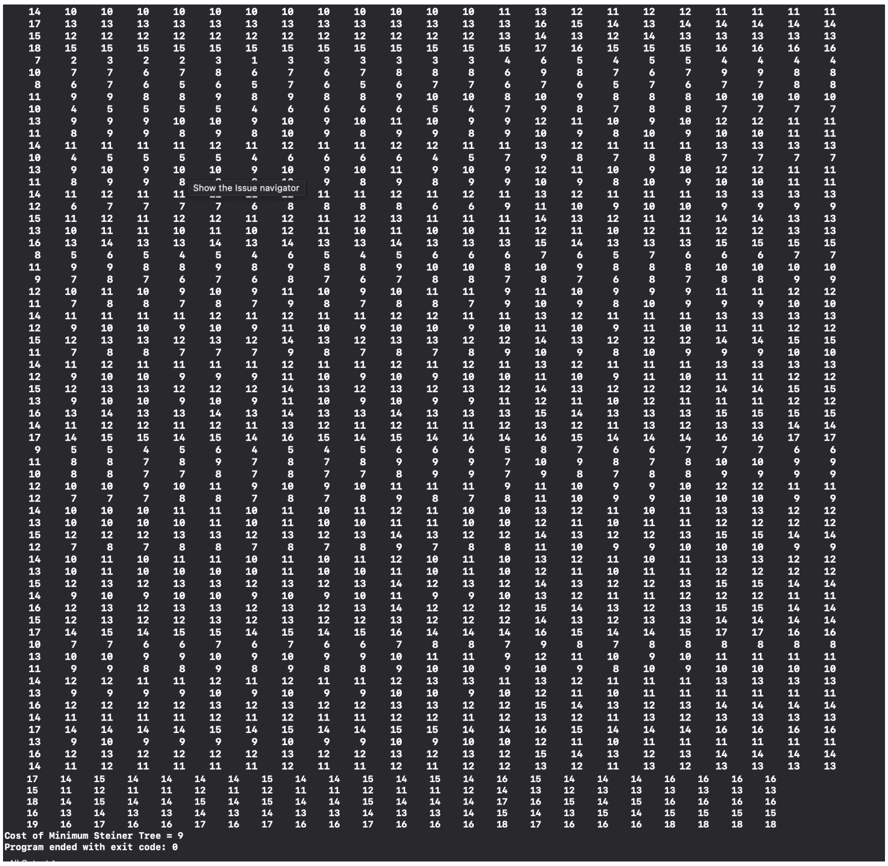
The graph is ready for algorithm :)

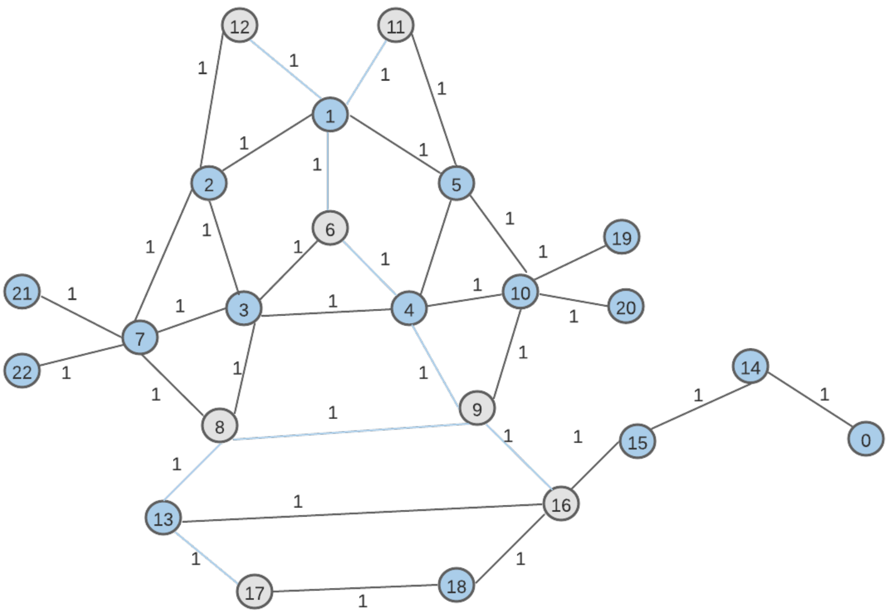
---------------------All Pair Shortest Path Matrix---------------------

0	7	7	6	5	6	6	6	5	4	5	7	8	4	1	2	3	5	4	6	6	7	7	7	6	5	8	9	4	6
7	0	1	2	2	1	1	2	3	3	2	1	1	4	6	5	4	5	5	3	3	3	3	2	4	4	2	2	5	6
7	1	0	1	2	2	2	1	2	3	3	2	1	3	6	5	4	4	5	4	4	2	2	3	3	4	3	2	5	5
6	2	1	0	1	2	1	1	1	2	2	3	2	2	5	4	3	3	4	3	3	2	2	2	2	3	4	3	4	4
5	2	2	1	0	1	1	2	2	1	1	2	3	3	4	3	2	4	3	2	2	3	3	2	3	2	3	4	3	5
6	1	2	2	1	0	2	3	3	2	1	1	2	4	5	4	3	5	4	2	2	4	4	3	4	3	2	3	4	6
6	1	2	1	1	2	0	2	2	2	2	2	2	3	5	4	3	4	4	3	3	3	3	1	3	3	3	3	4	5
6	2	1	1	2	3	2	0	1	2	3	3	2	2	5	4	3	3	4	4	4	1	1	3	2	3	4	3	4	4
5	3	2	1	2	3	2	1	0	1	2	4	3	1	4	3	2	2	3	3	3	2	2	3	1	2	5	4	3	3
4	3	3	2	1	2	2	2	1	0	1	3	4	2	3	2	1	3	2	2	2	3	3	3	2	1	4	5	2	4
5	2	3	2	1	1	2	3	2	1	0	2	3	3	4	3	2	4	3	1	1	4	4	3	3	2	3	4	3	5
7	1	2	3	2	1	2	3	4	3	2	0	2	5	6	5	4	6	5	3	3	4	4	3	5	4	1	3	5	7
8	1	1	2	3	2	2	2	3	4	3	2	0	4	7	6	5	5	6	4	4	3	3	3	4	5	3	1	6	6
4	4	3	2	3	4	3	2	1	2	3	5	4	0	3	2	1	1	2	4	4	3	3	4	2	3	6	5	2	2
1	6	6	5	4	5	5	5	4	3	4	6	7	3	0	1	2	4	3	5	5	6	6	6	5	4	7	8	3	5
2	5	5	4	3	4	4	4	3	2	3	5	6	2	1	0	1	3	2	4	4	5	5	5	4	3	6	7	2	4
3	4	4	3	2	3	3	3	2	1	2	4	5	1	2	1	0	2	1	3	3	4	4	4	3	2	5	6	1	3
5	5	4	3	4	5	4	3	2	3	4	6	5	1	4	3	2	0	1	5	5	4	4	5	3	4	7	6	3	1
4	5	5	4	3	4	4	4	3	2	3	5	6	2	3	2	1	1	0	4	4	5	5	5	4	3	6	7	2	2
6	3	4	3	2	2	3	4	3	2	1	3	4	4	5	4	3	5	4	0	2	5	5	4	4	3	4	5	4	6
6	3	4	3	2	2	3	4	3	2	1	3	4	4	5	4	3	5	4	2	0	5	5	4	4	3	4	5	4	6
7	3	2	2	3	4	3	1	2	3	4	4	3	3	6	5	4	4	5	5	5	0	2	4	3	4	5	4	5	5
7	3	2	2	3	4	3	1	2	3	4	4	3	3	6	5	4	4	5	5	5	2	0	4	3	4	5	4	5	5
7	2	3	2	2	3	1	3	3	3	3	3	3	4	6	5	4	5	5	4	4	4	4	0	4	4	4	4	5	6
6	4	3	2	3	4	3	2	1	2	3	5	4	2	5	4	3	3	4	4	4	3	3	4	0	3	6	5	4	4
5	4	4	3	2	3	3	3	2	1	2	4	5	3	4	3	2	4	3	3	3	4	4	4	3	0	5	6	3	5
8	2	3	4	3	2	3	4	5	4	3	1	3	6	7	6	5	7	6	4	4	5	5	4	6	5	0	4	6	8
9	2	2	3	4	3	3	3	4	5	4	3	1	5	8	7	6	6	7	5	5	4	4	4	5	6	4	0	7	7
4	5	5	4	3	4	4	4	3	2	3	5	6	2	3	2	1	3	2	4	4	5	5	5	4	3	6	7	0	4
6	6	5	4	5	6	5	4	3	4	5	7	6	2	5	4	3	1	2	6	6	5	5	6	4	5	8	7	4	0

------------------------------DP Table------------------------------

INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
6	6	5	4	5	6	5	4	3	4	5	7	6	2	5	4	3	1	2	6	6	5	5
4	5	5	4	3	4	4	4	3	2	3	5	6	2	3	2	1	3	2	4	4	5	5
7	8	7	6	6	7	7	6	5	5	6	8	8	4	6	5	4	4	4	7	7	7	7
9	2	2	3	4	3	3	3	4	5	4	3	1	5	8	7	6	6	7	5	5	4	4
11	8	7	7	8	9	8	7	7	8	9	9	7	7	10	9	8	7	8	10	10	8	8
10	7	7	7	7	7	7	7	7	7	7	8	7	7	9	8	7	8	8	8	8	8	8
12	10	9	9	10	10	10	9	9	10	10	11	9	9	11	10	9	9	10	11	11	10	10
8	2	3	4	3	2	3	4	5	4	3	1	3	6	7	6	5	7	6	4	4	5	5
11	8	8	8	8	8	8	8	8	8	8	8	9	8	10	9	8	8	8	9	9	9	9
9	7	8	7	6	6	7	8	7	6	6	6	8	7	8	7	6	8	7	7	7	9	9
12	10	10	10	9	9	10	10	10	9	9	9	11	9	11	10	9	9	9	10	10	11	11
11	4	5	6	6	[illegible]	[illegible]	[illegible]	7	7	6	4	4	8	10	9	8	9	9	7	7	7	7
14	10	10	10	11	11	10	10	10	11	11	10	10	10	13	12	11	10	11	12	12	11	11
12	9	10	10	9	9	9	10	10	9	9	9	9	10	11	10	9	11	10	10	10	11	11
15	12	12	12	12	12	12	12	12	12	12	12	12	12	14	13	12	12	12	13	13	13	13
5	4	4	3	2	3	3	3	2	1	2	4	5	3	4	3	2	4	3	3	3	4	4
8	8	7	6	6	7	7	6	5	5	6	8	8	5	7	6	5	5	5	7	7	7	7
6	6	6	5	4	5	5	5	4	3	4	6	7	4	5	4	3	5	4	5	5	6	6
9	9	9	8	7	8	8	8	7	6	7	9	10	6	8	7	6	6	6	8	8	9	9
10	6	6	6	6	6	6	6	6	6	6	7	6	7	9	8	7	8	8	7	7	7	7
13	10	9	9	10	10	10	9	9	9	10	11	9	9	12	11	10	9	10	11	11	10	10
11	8	8	8	8	8	8	8	8	8	8	9	8	9	10	9	8	10	9	9	9	9	9
14	11	11	11	11	11	11	11	11	11	11	12	11	11	13	12	11	11	11	12	12	12	12
9	6	7	6	5	5	6	7	6	5	5	5	7	7	8	7	6	8	7	6	6	8	8
12	10	10	10	9	9	10	10	9	9	9	9	11	9	11	10	9	9	9	10	10	11	11
10	8	9	8	7	7	8	9	8	7	7	7	9	8	9	8	7	9	8	8	8	10	10
13	11	12	11	10	10	11	12	11	10	10	10	12	10	12	11	10	10	10	11	11	13	13
12	8	9	9	8	8	8	9	9	8	8	8	8	10	11	10	9	11	10	9	9	10	10
15	12	12	12	12	12	12	12	12	12	12	12	12	12	14	13	12	12	12	13	13	13	13
13	10	11	11	10	10	10	11	11	10	10	10	10	11	12	11	10	12	11	11	11	12	12
16	13	14	14	13	13	13	14	14	13	13	13	13	13	15	14	13	13	13	14	14	15	15
6	4	3	2	3	4	3	2	1	2	3	5	4	2	5	4	3	3	4	4	4	3	3
8	7	6	5	6	7	6	5	4	5	6	8	7	4	7	6	5	4	5	7	7	6	6
7	7	6	5	5	6	6	5	4	4	5	7	7	4	6	5	4	5	5	6	6	6	6
9	9	8	7	8	9	8	7	6	7	8	10	9	6	8	7	6	6	7	9	9	8	8
10	6	5	5	6	7	6	5	5	6	7	7	5	6	9	8	7	7	8	8	8	6	6
12	9	8	8	9	10	9	8	8	9	10	10	8	8	11	10	9	8	9	11	11	9	9
11	9	8	8	9	9	9	8	8	8	9	10	8	8	10	9	8	9	9	10	10	9	9
13	11	10	10	11	12	11	10	10	11	12	12	10	10	12	11	10	10	11	13	13	11	11
10	6	6	6	6	6	6	6	6	6	6	6	7	7	9	8	7	8	8	7	7	7	7
13	9	9	9	9	9	9	9	9	9	9	9	10	9	12	11	10	9	10	10	10	10	10
11	9	9	9	8	8	9	9	8	8	8	8	10	9	10	9	8	10	9	9	9	10	10
14	11	11	11	11	11	11	11	11	11	11	11	12	11	13	12	11	11	11	12	12	12	12
13	8	8	8	9	9	8	8	8	9	9	8	8	9	12	11	10	10	11	10	10	9	9
15	11	11	11	12	12	11	11	11	12	12	11	11	11	14	13	12	11	12	13	13	12	12
14	11	11	11	11	11	11	11	11	11	11	11	11	11	13	12	11	12	12	12	12	12	12
16	13	13	13	14	14	13	13	13	14	14	13	13	13	15	14	13	13	14	15	15	14	14
7	6	5	4	4	5	5	4	3	3	4	6	6	4	6	5	4	5	5	5	5	5	5
10	9	8	7	7	8	8	7	6	6	7	9	9	6	9	8	7	6	7	8	8	8	8
8	8	7	6	6	7	7	6	5	5	6	8	8	6	7	6	5	7	6	7	7	7	7
11	11	10	9	9	10	10	9	8	8	9	11	11	8	10	9	8	8	8	10	10	10	10
11	8	7	7	8	8	8	7	7	7	8	9	7	8	10	9	8	9	9	9	9	8	8
14	11	10	10	11	11	11	10	10	10	11	12	10	10	13	12	11	10	11	12	12	11	11
12	10	9	9	10	10	10	9	9	9	10	11	9	10	11	10	9	11	10	11	11	10	10
15	13	12	12	13	13	13	12	12	12	13	14	12	12	14	13	12	12	12	14	14	13	13
11	8	8	8	7	7	8	8	7	7	7	7	9	8	10	9	8	9	9	8	8	9	9
14	11	11	11	10	10	11	11	10	10	10	10	12	10	13	12	11	10	11	11	11	12	12
12	10	10	10	9	9	10	10	9	9	9	9	11	10	11	10	9	11	10	10	10	11	11
15	13	13	13	12	12	13	13	12	12	12	12	14	12	14	13	12	12	12	13	13	14	14
14	10	10	10	10	10	10	10	10	10	10	10	10	11	13	12	11	12	12	11	11	11	11


Show the Issue navigator
Cost of Minimum Steiner Tree = 9
Program ended with exit code: 0


12
11
1
2
5
6
19
21
3
4
10
20
7
22
14
8
9
15
0
13
16
17
18