CSCE 5150 – Analysis of Computer Algorithms

Programming Assignment 4 – Graph Algorithms

Neha Goud Baddam

11519516 ¶

In [46]:
```
def prims(N,G):
    #importing numpy library
    import numpy as np
    #initializing variables
    Infinite = 9999999
    Output = np.zeros((N,), dtype=int)
    edge_count = 0

    #first vertex will be part of the output by default
    Output[0] = True

    print("\nMinimum Spanning Tree (MST) for given undirected graph G (V, E) with
    print("\n Vertex path : Edge Weight")

    while (edge_count < N - 1):
        minimum = Infinite
        idx1 = 0
        idx2 = 0
        for i in range(N):
            if Output[i]:
                for j in range(N):
                    if ((not Output[j]) and G[i][j]):
                        if minimum > G[i][j]:
                            minimum = G[i][j]
                            idx1 = i
                            idx2 = j
        print("\n V" + str(idx1+1) + " to " + "V" + str(idx2+1) + "    : " + str(
        Output[idx2] = True
        edge_count = edge_count+1

#function to take graph input
def graph_input():
    import numpy as np

    N = int(input("Enter number of edges in the graph: "))
    G = np.full((N,N), -1)
    #takes edge weights between each vertex in the graph
    for i in range (N):
        for j in range(N):
            if i != j and G[i][j] == -1:
                print("Enter weight between vertex",i+1,"and",j+1)
                G[i][j] =int(input())
                G[j][i] = G[i][j]
            elif i == j:
                G[i][j] = 0

    print("\nMatrix for the input graph is: \n",G)

    prims(N,G)
```

In [47]:
```
graph_input()
```

```
Enter number of edges in the graph: 8
Enter weight between vertex 1 and 2
6
Enter weight between vertex 1 and 3
14
Enter weight between vertex 1 and 4
8
Enter weight between vertex 1 and 5
5
Enter weight between vertex 1 and 6
0
Enter weight between vertex 1 and 7
0
Enter weight between vertex 1 and 8
0
Enter weight between vertex 2 and 3
0
Enter weight between vertex 2 and 4
0
Enter weight between vertex 2 and 5
12
Enter weight between vertex 2 and 6
0
Enter weight between vertex 2 and 7
0
Enter weight between vertex 2 and 8
0
Enter weight between vertex 3 and 4
3
Enter weight between vertex 3 and 5
0
Enter weight between vertex 3 and 6
0
Enter weight between vertex 3 and 7
0
Enter weight between vertex 3 and 8
0
Enter weight between vertex 4 and 5
0
Enter weight between vertex 4 and 6
10
Enter weight between vertex 4 and 7
0
Enter weight between vertex 4 and 8
0
Enter weight between vertex 5 and 6
7
Enter weight between vertex 5 and 7
9
Enter weight between vertex 5 and 8
0
Enter weight between vertex 6 and 7
0
Enter weight between vertex 6 and 8
```

```
15
Enter weight between vertex 7 and 8
0

Matrix for the input graph is:
 [[ 0  6 14  8  5  0  0  0]
 [ 6  0  0  0 12  0  0  0]
 [14  0  0  3  0  0  0  0]
 [ 8  0  3  0  0 10  0  0]
 [ 5 12  0  0  0  7  9  0]
 [ 0  0  0 10  7  0  0 15]
 [ 0  0  0  0  9  0  0  0]
 [ 0  0  0  0  0 15  0  0]]

Minimum Spanning Tree (MST) for given undirected graph G (V, E) with weights i
s:

 Vertex path : Edge Weight

 V1 to V5    : 5

 V1 to V2    : 6

 V5 to V6    : 7

 V1 to V4    : 8

 V4 to V3    : 3

 V5 to V7    : 9

 V6 to V8    : 15
```

In [48]:
```
graph_input()
```

```
Enter number of edges in the graph: 6
Enter weight between vertex 1 and 2
6
Enter weight between vertex 1 and 3
3
Enter weight between vertex 1 and 4
0
Enter weight between vertex 1 and 5
7
Enter weight between vertex 1 and 6
0
Enter weight between vertex 2 and 3
4
Enter weight between vertex 2 and 4
2
Enter weight between vertex 2 and 5
0
Enter weight between vertex 2 and 6
5
Enter weight between vertex 3 and 4
3
Enter weight between vertex 3 and 5
8
Enter weight between vertex 3 and 6
0
Enter weight between vertex 4 and 5
0
Enter weight between vertex 4 and 6
2
Enter weight between vertex 5 and 6
0

Matrix for the input graph is:
 [[0 6 3 0 7 0]
 [6 0 4 2 0 5]
 [3 4 0 3 8 0]
 [0 2 3 0 0 2]
 [7 0 8 0 0 0]
 [0 5 0 2 0 0]]

Minimum Spanning Tree (MST) for given undirected graph G (V, E) with weights i
s:

 Vertex path : Edge Weight

 V1 to V3    : 3

 V3 to V4    : 3

 V4 to V2    : 2

 V4 to V6    : 2

 V1 to V5    : 7
```

In [49]:
```
graph_input()
```

```
Enter number of edges in the graph: 7
Enter weight between vertex 1 and 2
28
Enter weight between vertex 1 and 3
0
Enter weight between vertex 1 and 4
0
Enter weight between vertex 1 and 5
0
Enter weight between vertex 1 and 6
10
Enter weight between vertex 1 and 7
0
Enter weight between vertex 2 and 3
16
Enter weight between vertex 2 and 4
0
Enter weight between vertex 2 and 5
0
Enter weight between vertex 2 and 6
0
Enter weight between vertex 2 and 7
14
Enter weight between vertex 3 and 4
12
Enter weight between vertex 3 and 5
0
Enter weight between vertex 3 and 6
0
Enter weight between vertex 3 and 7
0
Enter weight between vertex 4 and 5
22
Enter weight between vertex 4 and 6
0
Enter weight between vertex 4 and 7
18
Enter weight between vertex 5 and 6
25
Enter weight between vertex 5 and 7
24
Enter weight between vertex 6 and 7
0

Matrix for the input graph is:
 [[ 0 28  0  0  0 10  0]
 [28  0 16  0  0  0 14]
 [ 0 16  0 12  0  0  0]
 [ 0  0 12  0 22  0 18]
 [ 0  0  0 22  0 25 24]
 [10  0  0  0 25  0  0]
 [ 0 14  0 18 24  0  0]]

Minimum Spanning Tree (MST) for given undirected graph G (V, E) with weights
```

```
is:

 Vertex path : Edge Weight

 V1 to V6    : 10

 V6 to V5    : 25

 V5 to V4    : 22

 V4 to V3    : 12

 V3 to V2    : 16

 V2 to V7    : 14
```

In [ ]:

In [ ]: