Name: Ganesh Karigar

USN: 2VX23UE015

Expt_No_09: Prim's Algorithm

```
In [5]:
           2 graph = [[0, 28, 0, 0, 0, 10, 0],
                      [28, 0, 16, 0, 0, 0, 14],
[0, 16, 0, 12, 0, 0, 0],
           3
          4
           5
                      [0, 0, 12, 0, 22, 0, 18],
           6
                      [0, 0, 0, 22, 0, 0, 25, 24],
                      [10, 0, 0, 0, 25, 0, 0],
          8
                      [0, 14, 0, 18, 24, 0, 0]]
          9
          10 def get_min_key_vertex(key, in_mst):
          11
                  return min((v for v in range(len(key)) if not in_mst[v]), key=lambda v: key[v], default=-1)
         12
         13 def Prim_MCST(graph):
                 vertices = len(graph)
         14
                  parent = [-1] * vertices
key = [float('inf')] * vertices
         15
          16
          17
                  in_mst = [False] * vertices
          18
          19
                  key[0] = 0
                  total_cost = 0
          20
          21
          22
                  for _ in range(vertices):
          23
                      u = min((v for v in range(vertices) if not in_mst[v]), key=lambda v: key[v], default=-1)
          24
                      if u == -1:
          25
                          break
          26
                      in_mst[u] = True
          27
          28
                      for v in range(vertices):
                           if graph[u][v] != 0 and not in_mst[v] and graph[u][v] < key[v]:</pre>
          29
                               key[v]= graph[u][v]
parent[v] = u
          30
          31
          32
          33
                  total_cost =sum(key[1:])
          34
          35
                  return total_cost
          36
          37 min_cost = Prim_MCST(graph)
         38 print("Minimum Cost of Spanning Tree using Prims Algorithm:", min_cost)
```

Minimum Cost of Spanning Tree using Prims Algorithm: 99

```
In [6]:
           2 graph = [[0, 82, 0, 0, 0, 10, 0],
                      [82, 0, 16, 0, 0, 0, 41],
          3
          4
                      [0, 61, 0, 21, 0, 0, 0],
                      [0, 0, 21, 0, 22, 0, 81],
           6
                      [0, 0, 0, 22, 0, 0, 52, 42],
                      [10, 0, 0, 0, 52, 0, 0],
          7
          8
                      [0, 41, 0, 81, 42, 0, 0]]
          9
          10 def get_min_key_vertex(key, in_mst):
          11
                  return min((v for v in range(len(key)) if not in_mst[v]), key=lambda v: key[v], default=-1)
         12
         def Prim_MCST(graph):
         14
                  vertices = len(graph)
                 parent = [-1] * vertices
key = [float('inf')] * vertices
in_mst = [False] * vertices
         15
          16
         17
          18
         19
                  key[0] = 0
          20
                  total_cost = 0
          21
          22
                  for _ in range(vertices):
                      u = min((v for v in range(vertices) if not in_mst[v]), key=lambda v: key[v], default=-1)
         23
         24
                      if u == -1:
          25
                          break
          26
                      in_mst[u] = True
          27
          28
                      for v in range(vertices):
                           if graph[u][v] != 0 and not in_mst[v] and graph[u][v] < key[v]:</pre>
         29
          30
                               \text{key}[v] = \text{graph}[u][v]
                               parent[v] = u
          31
          32
          33
                  total_cost =sum(key[1:])
          34
          35
                  return total_cost
          36
          37 min_cost = Prim_MCST(graph)
          38 print("Minimum Cost of Spanning Tree using Prims Algorithm:", min_cost)
```

Minimum Cost of Spanning Tree using Prims Algorithm: 198

In []: 1