Name: Ganesh Karigar

USN: 2VX23UE015

Expt_No_09: Prim's Algorithm

In [5]:

```python

graph = [[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, 0, 25, 24],
        [10, 0, 0, 0, 25, 0, 0],
        [0, 14, 0, 18, 24, 0, 0]]

def get_min_key_vertex(key, in_mst):
    return min((v for v in range(len(key)) if not in_mst[v]), key=lambda v: key[v], default=-1)

def Prim_MCST(graph):
    vertices = len(graph)
    parent = [-1] * vertices
    key = [float('inf')] * vertices
    in_mst = [False] * vertices

    key[0] = 0
    total_cost = 0

    for _ in range(vertices):
        u = min((v for v in range(vertices) if not in_mst[v]), key=lambda v: key[v], default=-1)
        if u == -1:
            break
        in_mst[u] = True

        for v in range(vertices):
            if graph[u][v] != 0 and not in_mst[v] and graph[u][v] < key[v]:
                key[v]= graph[u][v]
                parent[v] = u

    total_cost =sum(key[1:])

    return total_cost

min_cost = Prim_MCST(graph)
print("Minimum Cost of Spanning Tree using Prims Algorithm:", min_cost)

```

```
Minimum Cost of Spanning Tree using Prims Algorithm: 99
```

In [6]:

```
graph = [[0, 82, 0, 0, 0, 10, 0],
        [82, 0, 16, 0, 0, 0, 41],
        [0, 61, 0, 21, 0, 0, 0],
        [0, 0, 21, 0, 22, 0, 81],
        [0, 0, 0, 22, 0, 0, 52, 42],
        [10, 0, 0, 0, 52, 0, 0],
        [0, 41, 0, 81, 42, 0, 0]]

def get_min_key_vertex(key, in_mst):
    return min((v for v in range(len(key)) if not in_mst[v]), key=lambda v: key[v], default=-1)

def Prim_MCST(graph):
    vertices = len(graph)
    parent = [-1] * vertices
    key = [float('inf')] * vertices
    in_mst = [False] * vertices

    key[0] = 0
    total_cost = 0

    for _ in range(vertices):
        u = min((v for v in range(vertices) if not in_mst[v]), key=lambda v: key[v], default=-1)
        if u == -1:
            break
        in_mst[u] = True

        for v in range(vertices):
            if graph[u][v] != 0 and not in_mst[v] and graph[u][v] < key[v]:
                key[v]= graph[u][v]
                parent[v] = u

    total_cost =sum(key[1:])

    return total_cost

min_cost = Prim_MCST(graph)
print("Minimum Cost of Spanning Tree using Prims Algorithm:", min_cost)

```

```
Minimum Cost of Spanning Tree using Prims Algorithm: 198
```

In [ ]:

```

```