

Seat No.: T191094350 Name : Aryan Sirdesai

Problem Statement : Implement Greedy search algorithm for : IV. Prim's Minimal Spanning Tree Algorithm

```
In [1]: import sys

class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                       for row in range(vertices)]

    def printMST(self, parent):
        print("Edge \tWeight")
        for i in range(1, self.V):
            print(parent[i], "-", i, "\t", self.graph[i][parent[i]])

    def minKey(self, key, mstSet):
        min = sys.maxsize
        for v in range(self.V):
            if key[v] < min and mstSet[v] == False:
                min = key[v]
                min_index = v
        return min_index

    def primMST(self):
        key = [sys.maxsize] * self.V
        parent = [None] * self.V
        key[0] = 0
        mstSet = [False] * self.V
        parent[0] = -1

        for cout in range(self.V):
            u = self.minKey(key, mstSet)
            mstSet[u] = True
            for v in range(self.V):
                if self.graph[u][v] > 0 and mstSet[v] == False and key[v] > self.g
                    key[v] = self.graph[u][v]
                    parent[v] = u

        self.printMST(parent)

if __name__ == '__main__':
    vertices = int(input("Enter the number of vertices: "))
    g = Graph(vertices)
    print("Enter the graph matrix:")
    for i in range(vertices):
        row = list(map(int, input().split()))
        for j in range(vertices):
            g.graph[i][j] = row[j]
    source_vertex = int(input("Enter the source vertex: "))
    g.primMST()
```

```
Enter the number of vertices: 5
Enter the graph matrix:
0 2 0 6 0
2 0 3 5 8
0 3 0 0 7
6 8 0 0 9
0 5 7 9 0
Enter the source vertex: 0
Edge      Weight
0 - 1      2
1 - 2      3
1 - 3      8
2 - 4      7
```

In []: