**Assignment 11**

**Name: Mudit Mohit**

**Enrolment No.: BT21GCS057**

**Section: B2**

**Question:** Implement the Greedy method (i.e........ Prim's algorithm) to find the minimum spanning tree of a given weighted undirected graph.

Ans.

import heapq

def prim\_mst(graph):

    n = len(graph)

    visited = [False] \* n

    pq = [(0, 0)]

    mst = [[] for \_ in range(n)]

    while pq:

        weight, u = heapq.heappop(pq)

        if visited[u]:

            continue

        visited[u] = True

        for v, w in graph[u]:

            if not visited[v]:

                heapq.heappush(pq, (w, v))

        if weight != 0:

            mst[u].append((weight, u))

            mst[v].append((weight, u))

    return mst

graph = [

        [(1, 4), (7, 8)],

        [(0, 4), (2, 8), (7, 11)],

        [(1, 8), (3, 7), (5, 4), (8, 2)],

        [(2, 7), (4, 9), (5, 14)],

        [(3, 9), (5, 10)],

        [(2, 4), (3, 14), (4, 10), (6, 2)],

        [(5, 2), (7, 1), (8, 6)],

        [(0, 8), (1, 11), (6, 1), (8, 7)],

        [(2, 2), (6, 6), (7, 7)]

        ]

mst = prim\_mst(graph)

print(mst)

Output:



Each element of the list represents a vertex, and the edges connecting it to its neighbors are stored as tuples containing the weight of the edge and the neighbor vertex.