Assignment : 7

Name: Shivraj Chaudar
Div: A **Class:** T.E
Roll no. : 3101018

#Code :

```
import heapq

def prim_mst(graph):
    n = len(graph)
    visited = [False] * n
    min_heap = [(0, 0)]
    mst_cost = 0
    edges = []

    while min_heap:
        cost, u = heapq.heappop(min_heap)
        if visited[u]:
            continue
        visited[u] = True
        mst_cost += cost

        for v, weight in graph[u]:
            if not visited[v]:
                heapq.heappush(min_heap, (weight, v))
                edges.append((u, v, weight))

    return mst_cost, edges

graph = [
    [(1, 2), (3, 6)],
    [(0, 2), (2, 3), (3, 8), (4, 5)],
    [(1, 3), (4, 7)],
    [(0, 6), (1, 8)],
    [(1, 5), (2, 7)]
]

mst_cost, mst_edges = prim_mst(graph)
print("Cost of the Minimum Spanning Tree:", mst_cost)
print("Edges in the Minimum Spanning Tree:", mst_edges)
```

#Output :

```
shivrajchaudar@Shivrajs-Macbook-Pro LP-II % /usr/local/bin/python /Users/Shivrajchaudar/
Desktop/LP-II/A7.py
Cost of the Minimum Spanning Tree: 16
Edges in the Minimum Spanning Tree: [(0, 1, 2), (0, 3, 6), (1, 2, 3), (1, 3, 8), (1, 4, 5), (2, 4, 7)]
shivrajchaudar@Shivrajs-Macbook-Pro LP-II %
```