

LP-II Assignment 7

Name: Tanaya Bhore

Roll no: 3101016

Class - TE A Batch - A

```
import heapq

def prim_mst(graph, start):
    num_vertices = len(graph)
    mst = []
    visited = [False] * num_vertices
    min_heap = []

    heapq.heappush(min_heap, (0, start, -1))

    while min_heap:
        weight, u, prev = heapq.heappop(min_heap)

        if visited[u]:
            continue

        visited[u] = True

        if prev != -1:
            mst.append((prev, u, weight))

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

    return mst

def get_graph_input():
    num_vertices = int(input("Enter the number of vertices: "))
    graph = {i: [] for i in range(num_vertices)}

    num_edges = int(input("Enter the number of edges: "))

    print("Enter the edges (start_vertex end_vertex weight):")
    for _ in range(num_edges):
        u, v, weight = map(int, input().split())
        graph[u].append((v, weight))
        graph[v].append((u, weight))

    return graph

def main():
    graph = get_graph_input()
```

```

start_vertex = int(input("Enter the starting vertex for Prim's algorithm: "))

mst = prim_mst(graph, start_vertex)

print("\nEdges in the Minimum Spanning Tree:")
for edge in mst:
    print(f"{edge[0]} - {edge[1]} with weight {edge[2]}")

if __name__ == "__main__":
    main()

```

OUTPUT:

```

student@student-OptiPlex-3020:~/Desktop$ python3 lp3.py
Enter the number of vertices: 4
Enter the number of edges: 5
Enter the edges (start_vertex end_vertex
weight): 0 1 2
0 2 3
1 2 2
1 3 4
2 3 4
Enter the starting vertex for Prim's algorithm: 0

Edges in the Minimum Spanning Tree:
0 - 1 with weight 2
1 - 2 with weight 2
1 - 3 with weight 4

```


