MST

```
graph = {}


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


for i in range(num_edges):
    while True:
        edge = input(f"Enter edge {i+1} in the format 'vertex1 vertex2 weight': ")
        edge = edge.split()
        if len(edge) == 3:
            break
        else:
            print("Invalid input, please try again.")
            continue


    vertex1 = edge[0]
    vertex2 = edge[1]
    weight = int(edge[2])


    if vertex1 not in graph:
        graph [vertex1] = {}
    if vertex2 not in graph:
        graph[vertex2] = {}
    graph[vertex1][vertex2] = weight
    graph[vertex2][vertex1] = weight


mst =[]
visited = set()
```

```
start_vertex = list(graph.keys())[0]
visited.add(start_vertex)

while len(visited) < num_vertices:
    min_edge = None
    for vertex in visited:
        for neighbor in graph [vertex]:
            if neighbor not in visited:
                if min_edge is None or graph [vertex][neighbor] < min_edge[2]:
                    min_edge = (vertex, neighbor, graph [vertex][neighbor])

    mst.append(min_edge)
    visited.add(min_edge[1])

print("Minimum Spanning Tree:")

for edge in mst:
    print(f"{edge[0]} - {edge[1]}; weight: {edge[2]}")
```