In [1]:

```python
import sys

def prim(graph):
    num_vertices = len(graph)
    # Initialize key values for all vertices to infinity
    key = [float('inf')] * num_vertices
    # Initialize parent array to store the MST
    parent = [-1] * num_vertices
    # Choose the first vertex as the starting point
    key[0] = 0
    mst_set = [False] * num_vertices

    for _ in range(num_vertices):
        # Find the vertex with the minimum key value from the set of vertice
        min_key = float('inf')
        min_index = -1
        for v in range(num_vertices):
            if not mst_set[v] and key[v] < min_key:
                min_key = key[v]
                min_index = v

        mst_set[min_index] = True

        # Update key values and parent for adjacent vertices of the chosen
        for v in range(num_vertices):
            if (
                graph[min_index][v] != 0  # Not in MST
                and not mst_set[v]  # Not already included in MST
                and graph[min_index][v] < key[v]  # Weight is less than the
            ):
                key[v] = graph[min_index][v]
                parent[v] = min_index

    return parent

def print_mst(graph, parent):
    print("Edge\tWeight")
    for i in range(1, len(graph)):
        print(f"{parent[i]} - {i}\t{graph[i][parent[i]]}")

# Example usage:
graph = [
    [0, 2, 0, 6, 0],
    [2, 0, 3, 8, 5],
    [0, 3, 0, 0, 7],
    [6, 8, 0, 0, 9],
    [0, 5, 7, 9, 0]
]

parent = prim(graph)
print_mst(graph, parent)
```

```
Edge    Weight
0 - 1   2
1 - 2   3
0 - 3   6
1 - 4   5
```