

## 96. Prims Algorithm

PROGRAM:-

```
import heapq

def prim(n, graph):
    mst = []
    mst_cost = 0
    visited = [False] * n
    min_heap = [(0, 0)] # (cost, vertex)

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

    return mst, mst_cost

# Example usage:
n = 4
graph = {
    0: [(1, 10), (2, 6), (3, 5)],
    1: [(0, 10), (3, 15)],
    2: [(0, 6), (3, 4)],
    3: [(0, 5), (1, 15), (2, 4)]
}

mst, mst_cost = prim(n, graph)
print("Edges in MST:", mst)
print("Total cost of MST:", mst_cost)
```

OUTPUT:-

```
Edges in MST: [(0, 1, 10), (0, 2, 6), (0, 3, 5), (3, 1, 15), (3, 2, 4)]
Total cost of MST: 19

=== Code Execution Successful ===
```

TIME COMPLEXITY:- $O(E \log V)$