

EX 13 IMPLEMENTATION OF PRIM'S ALGORITHM

```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>

#define V 5 // Number of vertices in the graph

int minKey(int key[], bool mstSet[]) {
    int min = INT_MAX, min_index;

    for (int v = 0; v < V; v++)
        if (mstSet[v] == false && key[v] < min)
            min = key[v], min_index = v;

    return min_index;
}

void printMST(int parent[], int graph[V][V]) {
    printf("Edge \tWeight\n");
    for (int i = 1; i < V; i++)
        printf("%d - %d \t%d \n", parent[i], i, graph[i][parent[i]]);
}

void primMST(int graph[V][V]) {
    int parent[V];
    int key[V];
    bool mstSet[V];

    for (int i = 0; i < V; i++)
        key[i] = INT_MAX, mstSet[i] = false;

    key[0] = 0;
    parent[0] = -1;

    for (int count = 0; count < V - 1; count++) {
        int u = minKey(key, mstSet);

        mstSet[u] = true;

        for (int v = 0; v < V; v++)
            if (graph[u][v] && mstSet[v] == false && graph[u][v] < key[v])
                parent[v] = u, key[v] = graph[u][v];
    }
}
```

```
    printMST(parent, graph);  
}
```

```
int main() {  
    int graph[V][V] = {  
        {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}  
    };  
  
    primMST(graph);  
  
    return 0;  
}
```

OUTPUT:

```
Input the number of vertices: 4  
Input the adjacency matrix for the graph:  
3  
4  
5  
66  
6  
5  
7  
8  
9  
1  
0  
1  
0  
1  
0  
1  
0  
1  
Edge    Weight  
0 - 1    6  
0 - 2    9  
2 - 3    0
```