

Experiment 6: To implement Prim's MST Algorithm using Greedy Method.

Code:

```
#include <stdio.h>

#include <limits.h>

#define MAX_VERTICES 100

int minKey(int key[], int mstSet[], int vertices) {
    int min = INT_MAX;
    int min_index = -1;
    for (int v = 0; v < vertices; ++v) {
        if (!mstSet[v] && key[v] < min) {
            min = key[v];
            min_index = v;
        }
    }
    return min_index;
}

void primMST(int graph[MAX_VERTICES][MAX_VERTICES], int vertices) {
    int parent[MAX_VERTICES];
    int key[MAX_VERTICES];
    int mstSet[MAX_VERTICES];
    for (int i = 0; i < vertices; ++i) {
        key[i] = INT_MAX;
        mstSet[i] = 0;
    }
```

```

key[0] = 0;

parent[0] = -1;

for (int count = 0; count < vertices - 1; ++count) {
    int u = minKey(key, mstSet, vertices);
    mstSet[u] = 1;
    for (int v = 0; v < vertices; ++v) {
        if (graph[u][v] && !mstSet[v] && graph[u][v] < key[v]) {
            parent[v] = u;
            key[v] = graph[u][v];
        }
    }
}

// Print the MST
printf("Edge \tWeight\n");

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

}

int main() {
    int vertices;

    printf("Enter the number of vertices: ");
    scanf("%d", &vertices);

    int graph[MAX_VERTICES][MAX_VERTICES];

    printf("Input the adjacency matrix for the graph:\n");

```

```

for (int i = 0; i < vertices; ++i) {
    for (int j = 0; j < vertices; ++j) {
        scanf("%d", &graph[i][j]);
    }
    primMST(graph, vertices);
    return 0;
}

```

Output:

```

Enter the number of vertices: 5
Input the adjacency matrix for the 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
Edge      Weight
0 - 1     2
1 - 2     3
0 - 3     6
1 - 4     5

```