

## exp@2: implementation of minimum spanning tree using prim's algorithm.

Aim: To implementation of minimum spanning tree using prim's algorithm

### Algorithm:

- \* Start.
- \* Input no.of vertices and adjacency matrix.
- \* Select an arbitrary vertex as starting point.
- \* Find the minimum edge connecting visited and unvisited nodes.
- \* Add edge to MST.
- \* Repeat until all vertices are included.
- \* Print MST and cost.
- \* Stop.

### Program:

```
#include <stdio.h>
#define V 5
#define INF 999
int main() {
    int g[V][V] = {
        {0, 2, 0, 6, 0}, // Row 0
        {2, 0, 3, 8, 1}, // Row 1
        {0, 3, 0, 0, 2}, // Row 2
        {6, 8, 0, 0, 4}, // Row 3
        {0, 1, 5, 7, 9} // Row 4
    };
    int key[V], vis[V] = {0}, parent[V], u;
    for (int i=0; i<V; i++) key[i] = INF;
    key[0] = 0; parent[0] = -1;
    for (int c=0; c<V-1; c++) {
        int min = INF;
        for (int i=0; i<V; i++)
            if (!vis[i] && key[i] < min) { min = key[i]; u = i; }
        vis[u] = 1;
        for (int v=0; v<V; v++)
            if (!vis[v] && g[u][v] < key[v]) key[v] = g[u][v], parent[v] = u;
    }
}
```

vis[v] = 1;

for (int v=0; v<V; v++)

{ if (g[w][v] && vis[v] == 0 && g[w][v] < key[v]) {

parent[v] = w;

key[v] = g[w][v];

}

3

printf("MST edges:\n");

for (int i=1; i<V; i++)

printf("%d - %d\n", parent[G], i);

4

### Output:

MST edges:

0 - 1

1 - 2

0 - 3

1 - 4

Result: Thus, the program executed successfully.