24. Prim’s Algorithm (Minimum Spanning Tree)

Aim:
To find Minimum Spanning Tree (MST) using Prim’s algorithm.

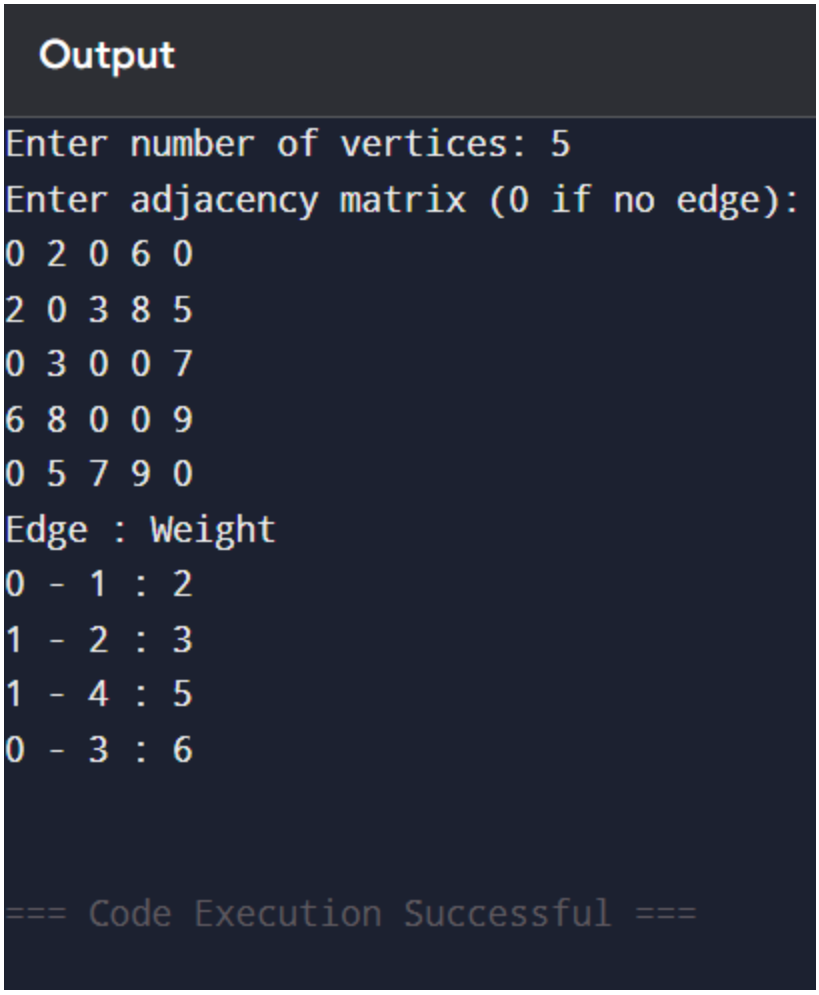
Algorithm:

1. Start from any vertex.
2. Pick the minimum edge that connects a visited vertex to an unvisited vertex.
3. Repeat until all vertices are included.

CODE:

```
#include <stdio.h>
#define INF 9999
#define MAX 20

void prim(int G[MAX][MAX], int n) {
    int selected[MAX], no_edge = 0;
    int x, y;
    for (int i = 0; i < n; i++) selected[i] = 0;
    selected[0] = 1;

    printf("Edge : Weight\n");
    while (no_edge < n - 1) {
        int min = INF;
        x = y = 0;
        for (int i = 0; i < n; i++) {
            if (selected[i]) {
                for (int j = 0; j < n; j++) {
                    if (!selected[j] && G[i][j]) {
                        if (min > G[i][j]) {
                            min = G[i][j];
                            x = i;
                            y = j;
                        }
```

```
                }
            }
        }
    }
    printf("%d - %d : %d\n", x, y, G[x][y]);
    selected[y] = 1;
    no_edge++;
  }
}

int main() {
  int n, G[MAX][MAX];
  printf("Enter number of vertices: ");
  scanf("%d", &n);
  printf("Enter adjacency matrix (0 if no edge):\n");
  for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
      scanf("%d", &G[i][j]);

  prim(G, n);
  return 0;
}
```



RESULT:

The program successfully executed and displayed the Prim's Algorithm.