Syeda Reeha Quasar

syedareehaquasar@gmail.com

Algorithm and Design Analysis Lab Internal

14114802719  
5C7  
Syeda Reeha Quasar

Prim’s Algorithm

Minimum Spanning Tree Algorithm

### AIM: To implement Prim’s Algorithm to find minimum Spanning Tree (MST).

# **Source Code**

#include <bits/stdc++.h>

using namespace std;

#define V 5

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;

}

// printing edges and weight

void printMST(int parent[], int graph[V][V])

{

cout << "Edge \tWeight\n";

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

cout << parent[i] << " - " << i << " \t" << graph[i][parent[i]] << " \n";

}

void prim(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, 9, 75, 0, 0},

{9, 0, 95, 19, 42},

{75, 95, 0, 51, 66},

{0, 19, 51, 0, 31},

{0, 42, 66, 31, 0}};

auto start = chrono::high\_resolution\_clock::now();

ios\_base::sync\_with\_stdio(false);

prim(graph);

auto end = chrono::high\_resolution\_clock::now();

double time\_taken = chrono::duration\_cast<chrono::nanoseconds>(end - start).count();

time\_taken \*= 1e-9 \* 1000;

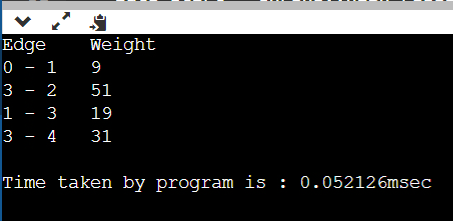
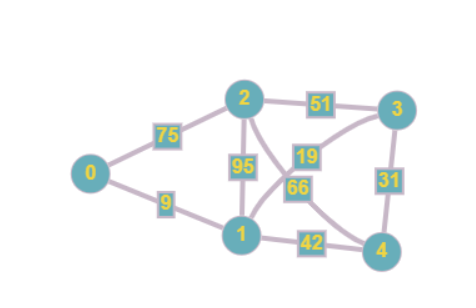
cout << "\nTime taken by program is : " << time\_taken << setprecision(6);

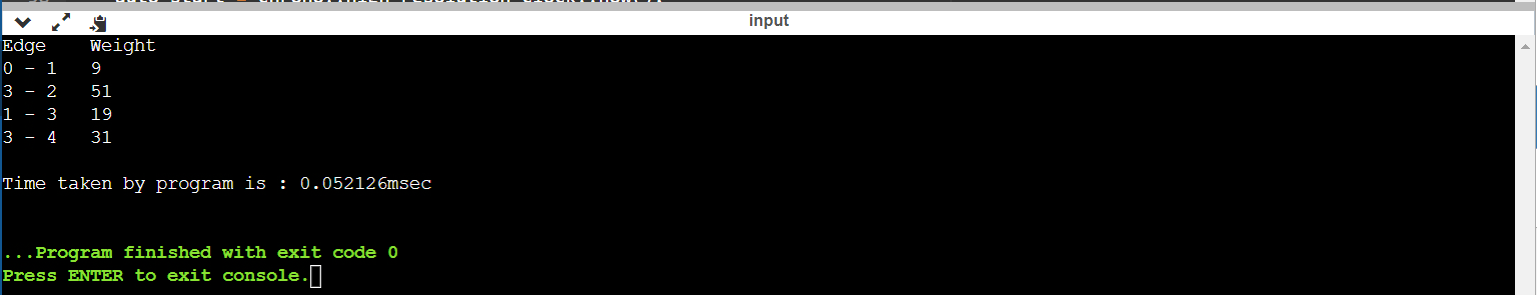
cout << "msec" << endl;

return 0;

}

# **Output**

****



## MST

**51**

**31**

**19**

**9**

**2**

**4**

**3**

**1**

**0**