Practical-10

Aim : implementation of prim's algorithm

Theory :

1). Create a set *mstSet* that keeps track of vertices already included in MST.

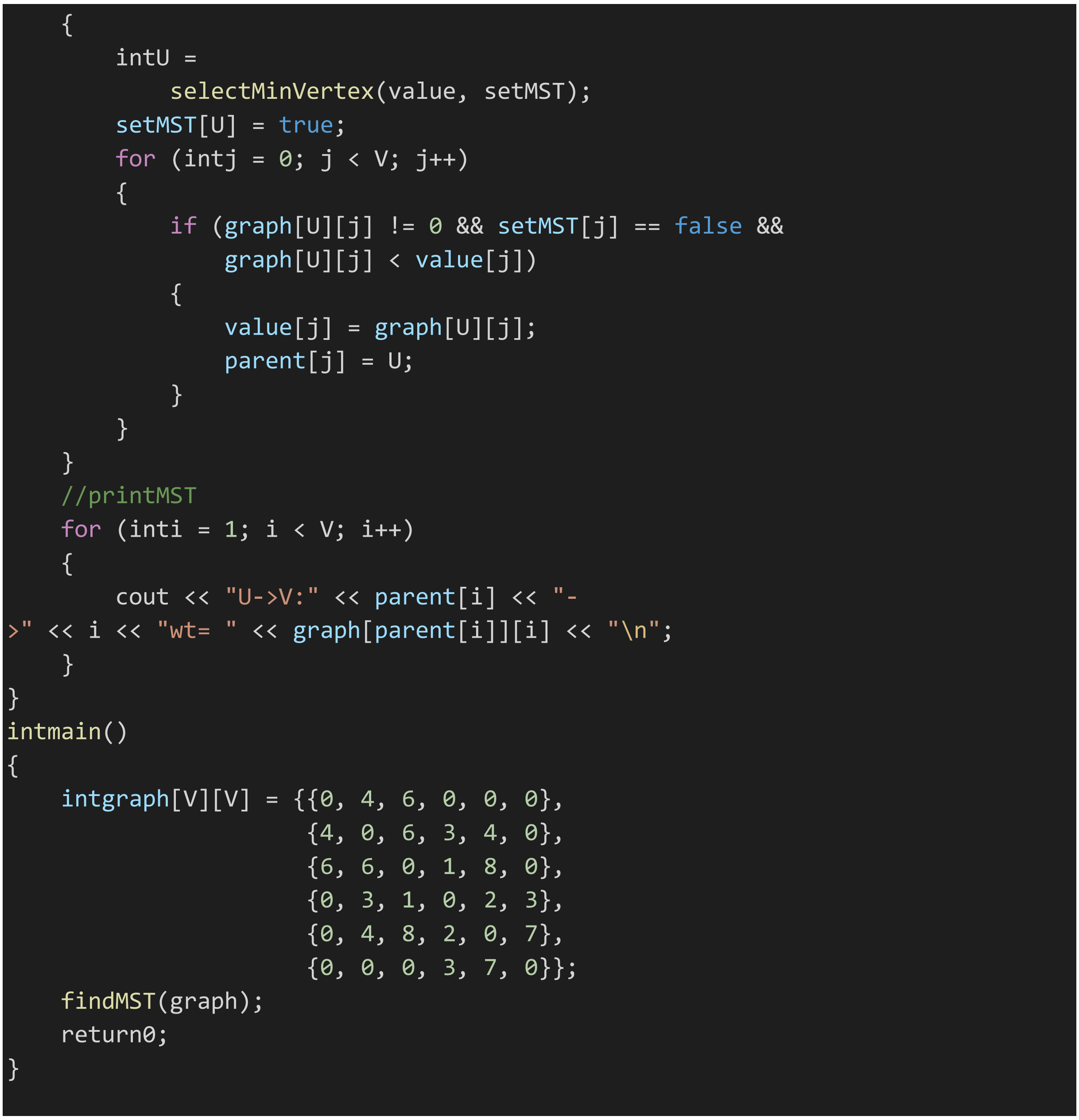
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.

3) While mstSet doesn’t include all vertices

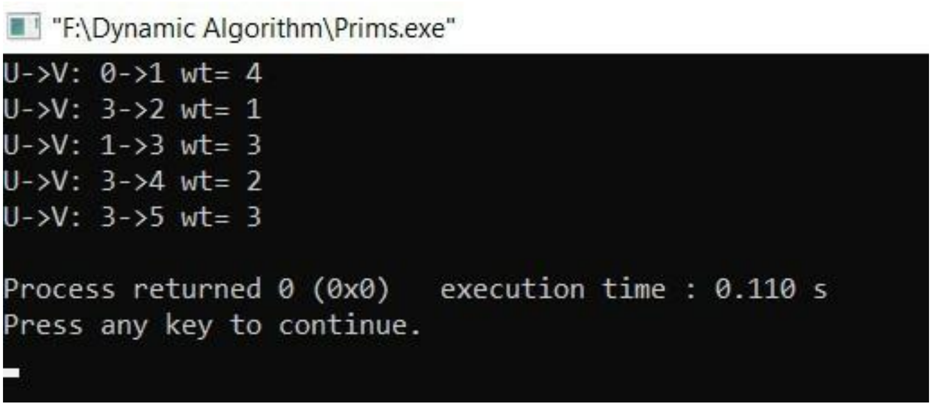
a) Pick a vertex *u* which is not there in *mstSet* and has minimum key value.
b) Include *u* to mstSet.
c) Update key value of all adjacent vertices of *u*. To update the key values, iterate through all adjacent vertices. For every adjacent vertex *v*, if weight of edge *u*-*v* is less than the previous key value of *v*, update the key value as weight of *u*-*v*

Code :

```
#include <bits/stdc++.h>
#defineV6
usingnamespacestd;
intselectMinVertex(vector<int> &value, vector<bool> &setMST)
{
    intminimum = INT_MAX;
    intvertex;
    for (inti = 0; i < V; i++)
    {
        if (setMST[i] == false && value[i] < minimum)
        {
            minimum = value[i];
            vertex = i;
        }
    }
    returnvertex;
}
voidfindMST(intgraph[V][V])
{
    intparent[V];
    vector<int> value(V, INT_MAX);
    vector<bool> setMST(V, false);
    value[0] = 0; //Assumingstartpointasnode-0
    parent[0] = -1;
    for (inti = 0; i < V - 1; i++)
```



```
    {
        intU =
            selectMinVertex(value, setMST);
        setMST[U] = true;
        for (intj = 0; j < V; j++)
        {
            if (graph[U][j] != 0 && setMST[j] == false &&
                graph[U][j] < value[j])
            {
                value[j] = graph[U][j];
                parent[j] = U;
            }
        }
    }
    //printMST
    for (inti = 1; i < V; i++)
    {
        cout << "U->V:" << parent[i] << "-
>" << i << "wt= " << graph[parent[i]][i] << "\n";
    }
}
intmain()
{
    intgraph[V][V] = {{0, 4, 6, 0, 0, 0},
                      {4, 0, 6, 3, 4, 0},
                      {6, 6, 0, 1, 8, 0},
                      {0, 3, 1, 0, 2, 3},
                      {0, 4, 8, 2, 0, 7},
                      {0, 0, 0, 3, 7, 0}};
    findMST(graph);
    return0;
}
```



"F:\Dynamic Algorithm\Prims.exe"

```
U->V: 0->1 wt= 4
U->V: 3->2 wt= 1
U->V: 1->3 wt= 3
U->V: 3->4 wt= 2
U->V: 3->5 wt= 3

Process returned 0 (0x0)   execution time : 0.110 s
Press any key to continue.
```