CS2023 - Inclass Lab
Week 11

Name: Wijetunga W.L.N.K Index Number: 200733D

GitHub Repository - https://github.com/namiwijeuom/CS2023-Data-Structures-and-Algorithms-In-class-Lab-Exercises/tree/main/Lab%2011

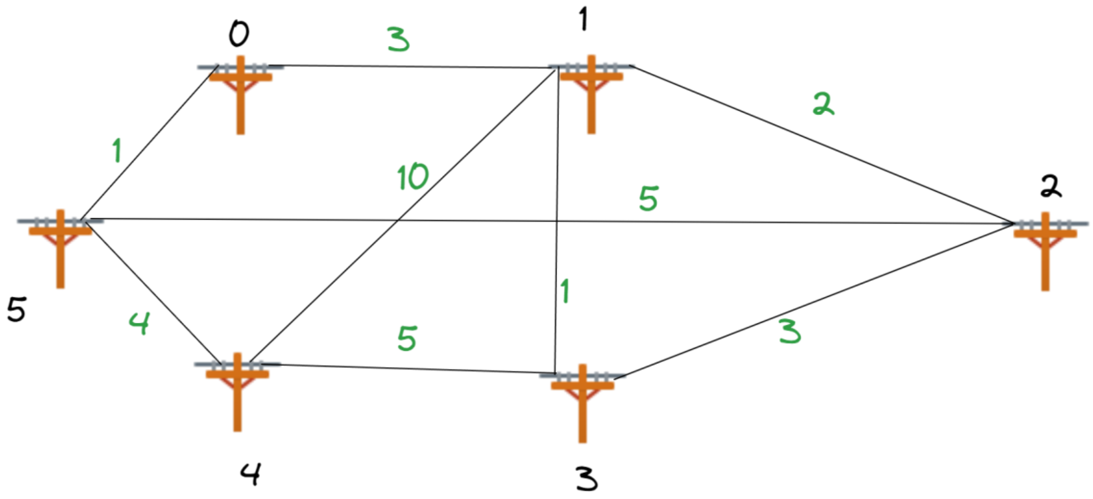
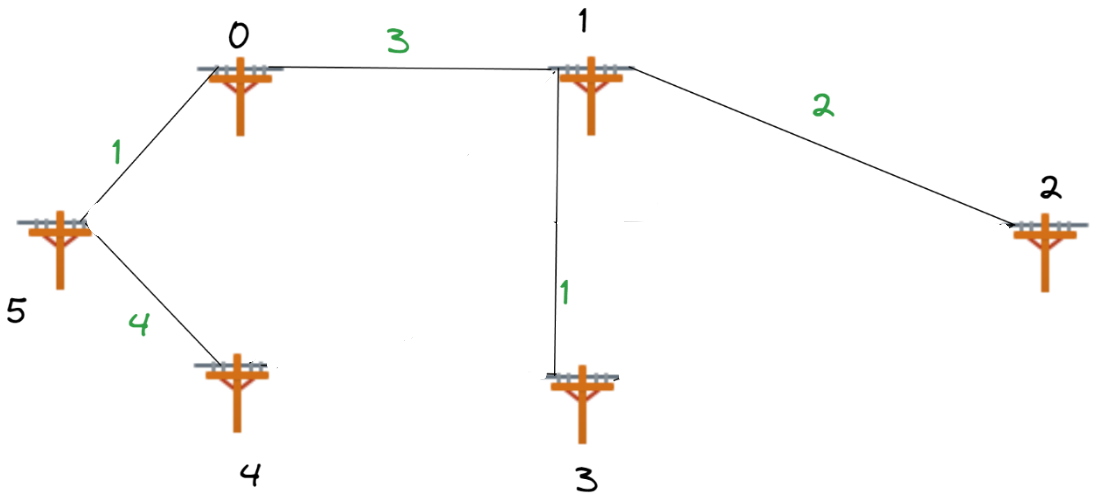


Figure 1: Telephone Pole Graph

1. Write the adjacency matrix for the graph on Fig 1.

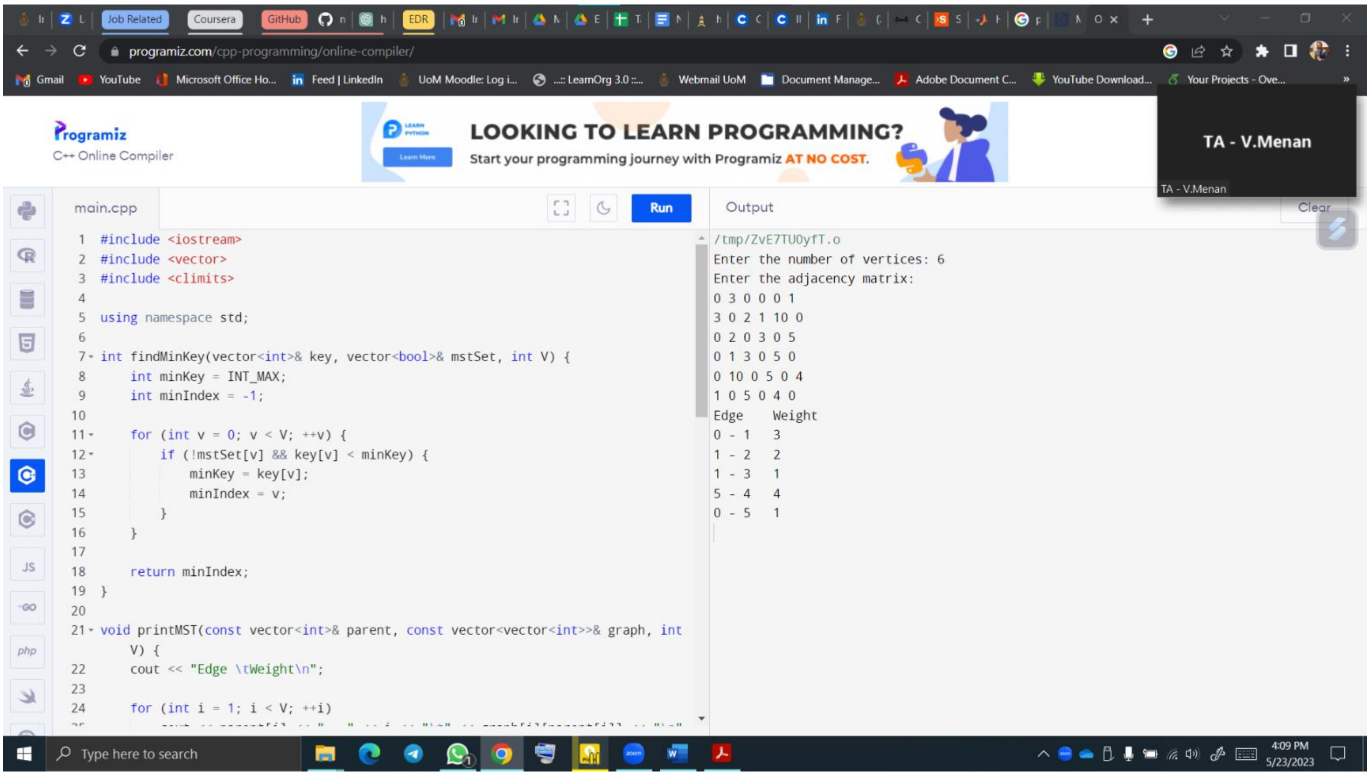
	0	1	2	3	4	5
0	0	3	0	0	0	1
1	3	0	2	1	10	0
2	0	2	0	3	0	5
3	0	1	3	0	5	0
4	0	10	0	5	0	4
5	1	0	5	0	4	0

2. Calculate and draw the minimum spanning tree for the graph in Fig1, taking **Node 3** as the start node.



Cost of the MST = 1 + 4 + 3 + 2 + 1 = 11

3. Implement Prim’s MST algorithm and obtain the minimum spanning tree taking **Node 0** as the start node. Take screen shot of your output.


programiz.com/cpp-programming/online-compiler/
Programiz
C++ Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
TA - V.Menan
main.cpp
Run
Output
Clear
Type here to search
4:09 PM
5/23/2023

```cpp
#include <iostream>
#include <vector>
#include <climits>

using namespace std;

int findMinKey(vector<int>& key, vector<bool>& mstSet, int V) {
    int minKey = INT_MAX;
    int minIndex = -1;

    for (int v = 0; v < V; ++v) {
        if (!mstSet[v] && key[v] < minKey) {
            minKey = key[v];
            minIndex = v;
        }
    }

    return minIndex;
}

void printMST(const vector<int>& parent, const vector<vector<int>>& graph, int
    V) {
    cout << "Edge \tWeight\n";

    for (int i = 1; i < V; ++i)
        cout << parent[i] << " - " << i << "\t" << graph[i][parent[i]] << "\n";
```

```
/tmp/ZvE7TUOyfT.o
Enter the number of vertices: 6
Enter the adjacency matrix:
0 3 0 0 0 1
3 0 2 1 10 0
0 2 0 3 0 5
0 1 3 0 5 0
0 10 0 5 0 4
1 0 5 0 4 0
Edge    Weight
0 - 1   3
1 - 2   2
1 - 3   1
5 - 4   4
0 - 5   1
```

4. Are the MSTs in Question 2 and Question 3 the same?

   Yes

   What is the condition for a graph to have only 1 minimum spanning tree?

   Answer - All the edge weights of the graph should be unique i.e. each edge of the graph should have a distinct weight.

5. Discuss the time complexity between Prim's Algorithm and Kruskal's Algorithm.

Prim's Algorithm	Kruskal's Algorithm
O(V^2) with adjacency matrix representation	**O(E log E)** or **O(E log V),** depending on the sorting algorithm used
O((V + E) log V) with adjacency list representation using a binary heap	For performing a disjoint set union-find operation for each edge, **O(1)** or a small constant value.
Time complexity depends on the data structure used to implement the priority queue to extract the minimum key value vertex.	The primary factor that affects the time complexity is the sorting of edges based on their weights.

Observations

- In most cases, Prim's algorithm has a higher time complexity than Kruskal's algorithm specially when the graph is dense.
- Time complexity of Kruskal's algorithm is depending on the sorting operation used.
- For sparse graphs, Kruskal's algorithm is efficient