

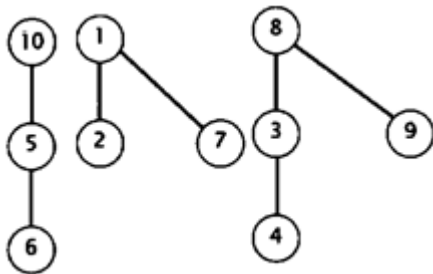
Heritage Institute of Technology
BTech (IoT) 2nd Year 2nd Semester Group 1/2
Paper: Design and Analysis of Algorithms Lab (CSEN 2251)
2024

Notes:

- Language: C
- Assume suitable function prototypes for the functions / operators mentioned.
- Provide source code listing with internal documentation/comments wherever applicable.
- Provide results with suitable test data / program (show on white page of lab workbook)
- Write your interpretation in case not obvious
- Write the question first

Day 9. Union-Find Algorithm and Kruskal's Minimum Spanning Tree

1. **[Connected Component Identification with Union-Find Algorithm]** Write a program that efficiently identifies the number of connected components within an undirected graph and delineates the nodes belonging to each connected component. Utilize the Union-Find algorithm for proficient component identification. Given an undirected graph with N nodes numbered from 1 to N and M edges, the graph may exhibit disconnection, resulting in multiple connected components. To aid in the implementation, the program should read input from a file named "adj.txt", adhering to the following structure:
 - a. The initial line contains the count of vertices in the graph.
 - b. The subsequent line contains the number of edges.
 - c. Subsequent lines consist of pairs of vertices separated by a single space, denoting an edge between them. For instance, "1 2" signifies that node 1 is linked to node 2, and vice versa, in an undirected graph.
 Additionally, the following utility functions need to be defined to facilitate implementation:
 - *void MAKESET(int s[], int size)* : Establishes new sets/trees containing individual elements/nodes.
 - *int FIND(int s[], int size, int x)* : Retrieves the identifier of the set/root of the tree to which node x pertains.
 - *void UNION(int s[], int size, int x, int y)* : Substitutes two sets encompassing nodes x and y with their amalgamation and updates the identifier of the set or parent of x as y.



adj.txt

```
10
7
1 2
1 7
3 4
3 8
8 9
5 6
5 10
```

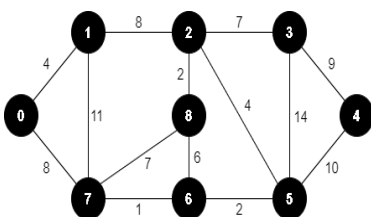
Output (Expected)

3 Connected components

{1, 2, 7}, {3, 4, 8, 9}, {5, 6, 10}

Your program should efficiently utilize the Union-Find algorithm to determine the number of connected components in the given graph and accurately identify the nodes belonging to each component.

2. **[Kruskal's Minimum Spanning Tree using UNION-FIND Algorithm]** Using UNION-FIND algorithm or its variation, write a program to apply Kruskal's algorithm to find the MST of a given weighted graph. The program should read input from a file named "adj.txt", with additional info of weight along with the edge information.

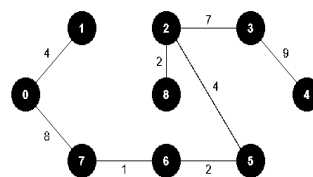


adj.txt

```
9
14
0 1 4
0 7 8
1 7 11
1 2 8
2 3 7
2 5 4
```

adj.txt...

```
2 8 2
3 4 9
3 5 14
4 5 10
5 6 2
6 7 1
6 8 6
7 8 7
```



Output (Expected)

Edges:

{6,7,1},{2,8,2},{5,6,2},{0,1,4},{2,5,4},

{2,3,7},{0,7,8},{3,4,9}

Total weight: 37