

CS 590 - Assignment 5

Submitted By: Shweta Madhale

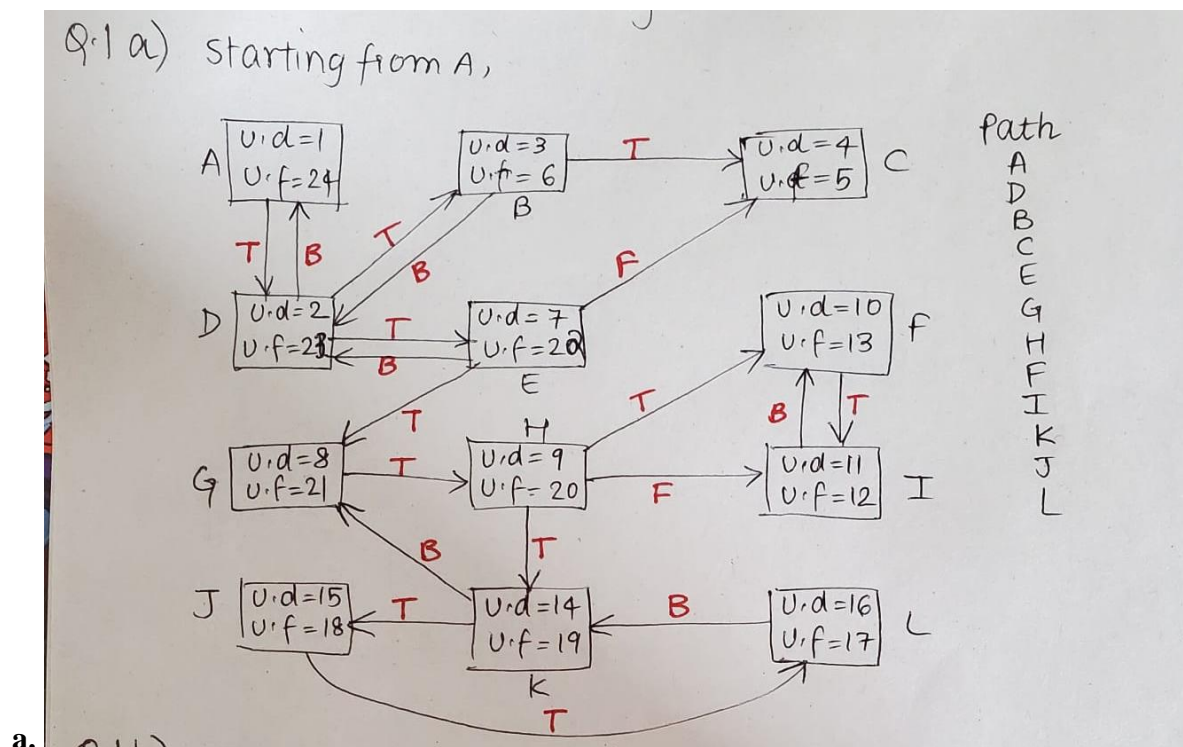
CWID: 20015921

Abstract:

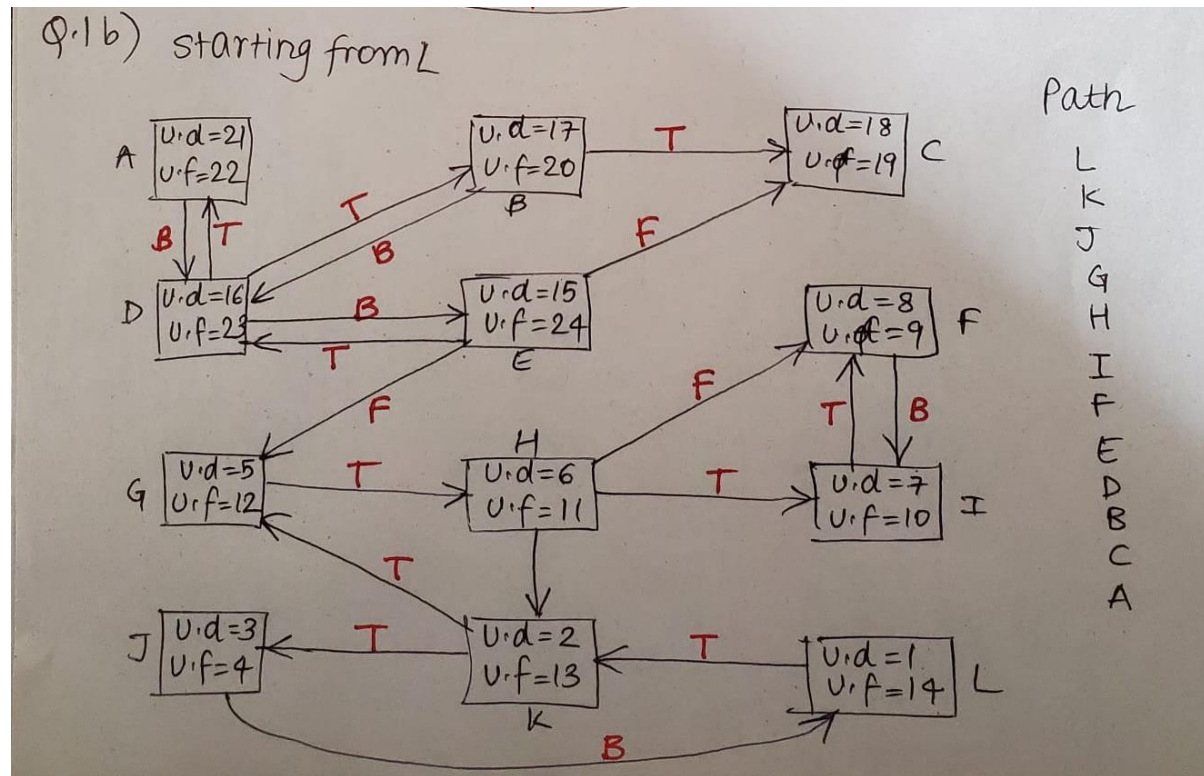
Depth first search (DFS) and Breadth First Search (BFS) are graph searching algorithms. The BFS algorithm expands the frontier across the breadth between the discovered and undiscovered vertices uniformly. It uses the concept of first-in, first-out for the nodes. It is used to find the shortest path distances. While the depth first search explores the edges of the most recently discovered vertex which has more undiscovered edges leaving it. It keeps timestamps for each vertex. It has two timestamps, for recording the first discovery time and the other for when search finishes examining the adjacency of that vertex. This algorithm is also used to find the shortest path distance. A few problems related to the DFS and BFS were solved, these algorithms were implemented in C++ for finding the shortest path starting from a source vertex. There are different types of edges viz tree, back, forward, and cross. These edges were identified in the problem. Also, the strongly connected components (SCC), equivalence classes of vertices which are mutually reachable, were also identified.

Observations:

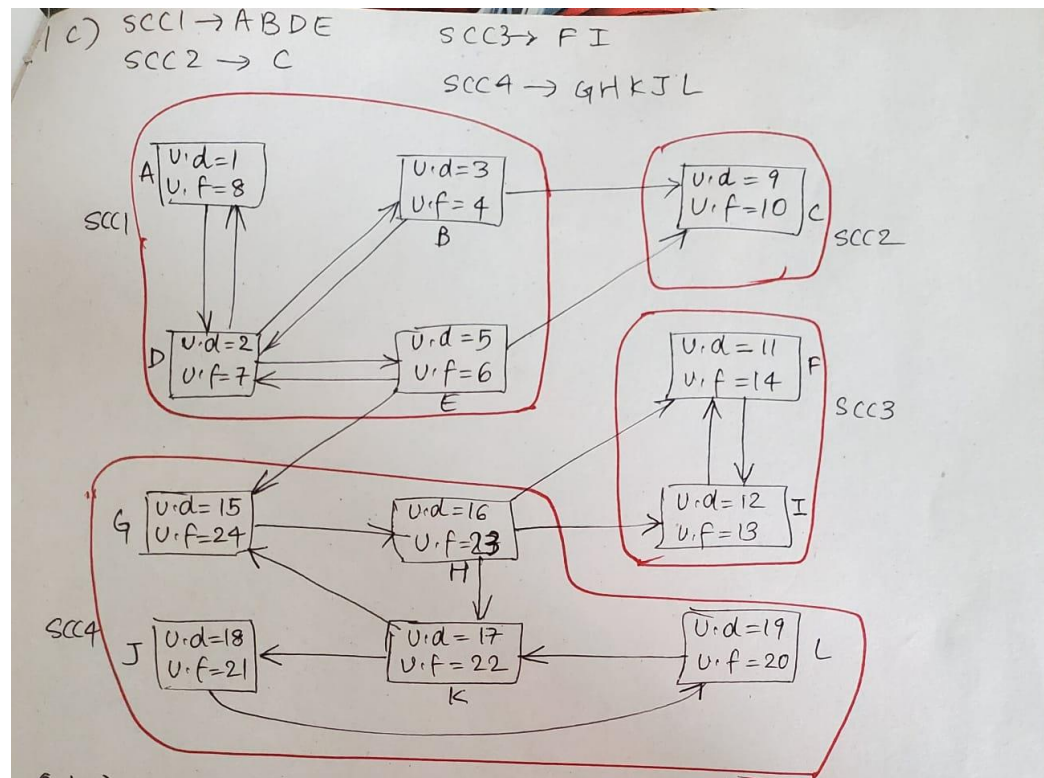
Q.1



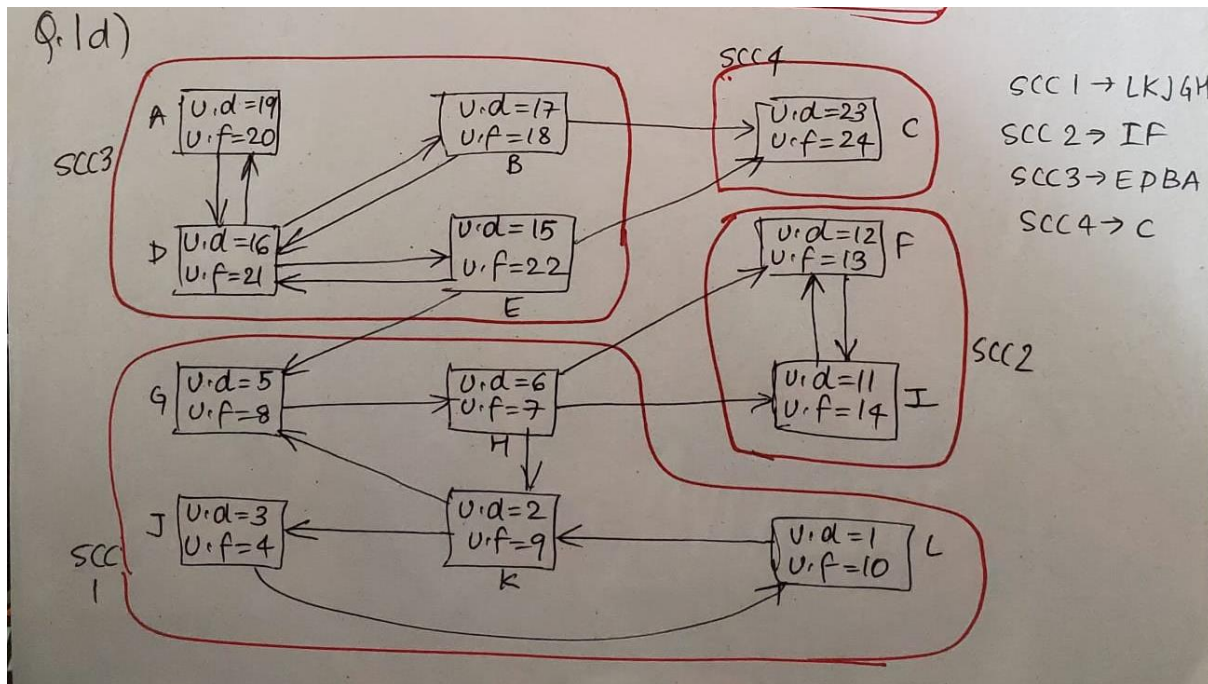
b.



c.



d.



For given edges and weights, the BFS and DFS provides following output

```

BFS output:
0 1 2 3 5 4 7 6
DFS output:
0 1 2 5 7 3 4 6 8 9
  
```

Analysis and Conclusion:

- The BFS algorithm works as a queue, it visits the node and adds additional undiscovered nodes originating from that node to the queue.
- The start node is added to the set of visited nodes, and its adjacent nodes are added to the queue and then visited later.
- The time complexity for BFS is $O(V+E)$ for V vertices and E edges.
- The depth first algorithm starts from a node and finds the adjacent nodes and continues for all nodes in the graph.
- The time complexity for DFS is also $O(V+E)$, for V vertices and E edges.