Data Structure and Algorithms

Lab11

Submitted to:	**Ms. Sehar Waqar**
Submitted by:	**SABA**
Roll no:	**2019-CE-04**
Date:	**13th July,2021**

Department of Computer Engineering

UET,Lahore

Question no 3:

a) Suppose you do BFS starting at the node A. What does BFS find as the shortest path from A to C?

Answer: **A-> C –>B**

b) If you take the weights into account, what is the shortest path from A to C?

Answer: **A-> B->C**

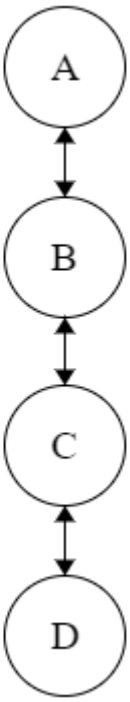
c) Try to think about how you would design an algorithm to find shortest paths in weighted graphs. Can you modify BFS to do it?

Answer: Yes, we can do it with BFS. We take two arguments in this case one is source and the other one is destination. Before enqueue, we check their weights and the distance of previous node. By keeping the distance of previous nodes and compare it with the next one we traverse the graph using bfs and stops where we find our destination.

Question no 4:

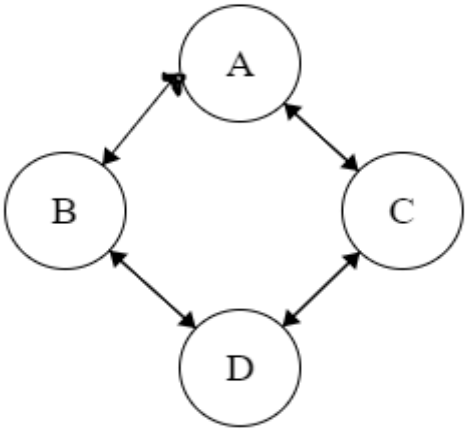
Give one example of a connected undirected graph on four vertices, A, B, C, and D, so that both DFS and BFS search discover the vertices in the same order when starting at vertex A. Then give one example of a connected undirected graph on four vertices, A, B, C, and D where BFS and DFS discover the vertices in a different order when started at A. Assume that both DFS and BFS iterate over neighbour's in alphabetical order.

Answer:



BFS : A,B,C,D

DFS : A,B,C,D



BFS : A,C,D,B

DFS : A,B,C,D

Question 5 :

a) Design an algorithm that verifies if the fuel distribution system has the following property:

Answer:

```
bool IsPathExists(int i, int j){
    if (i==j){
       return true; }
    Node *src = head;
    while(src->next != NULL){
        if(there is an adge between node i and node j){
            i = src->data;
            break;}
         src = src->next;
       }
    }
```

Running Time:

O (number of nodes visited + number of edges scanned) = O(m + n).

b) Design an algorithm that verifies if the fuel distribution system has the following property: No single pipe failure can isolate any node from either distributing fuel to the rest of the system or consuming fuel from the rest of the system. That is, removing a single edge cannot cause any node to be unreachable from any other node. Your algorithm should run in O(mn+m^2). You may assume an algorithm that satisfies the property in part (a) exists and use it directly.

Answer:

```
bool IsPathExists(int i, int j){
    if (i==j){
       return true; }
    Node *src = head;
    while(src->next != NULL){
        if(there is an adge between node i and node j){
            if(there is an adge between node i and a node next to j){
                i = src->data;
                break;}      }
         src = src->next;
       }
    }
```