CSCI-310, Data Structures

HW 10 – Shortest Paths

Modified BFS to Encode Shortest Paths

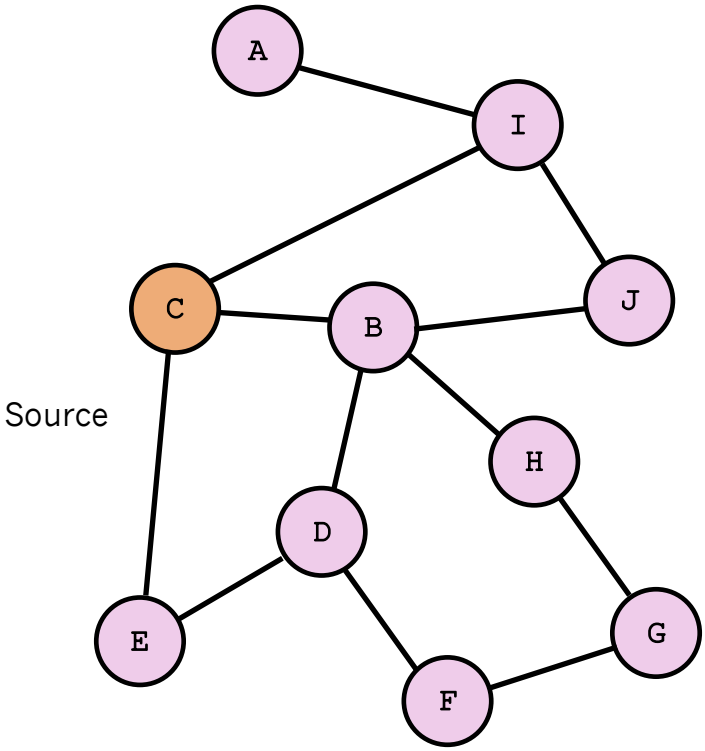
```
bfs(vertex v){
    Q = new Queue
    prev = empty list of vertices

    visit(v)
    Q.enqueue(v)

    while(queue is not empty) {
        u = Q.dequeue()
        for(each unvisited neighbor w of u){
            visit(w)
            prev[w] = u  // This line added to BFS
            Q.enqueue(w)
        }
    }
}
```

Recovering Shortest Paths

```
Shortest_path(vertex s, vertex t){
    prev = list of vertices from BFS
    sp = empty list of vertices

    current = t
    while (prev[current] has a vertex){
        add current to front of sp
        current = prev[current]
    }

    if (sp.length > 0){
        add s to front of sp
    }

    return sp
}
```

Problem 1 (5 points)

Execute by hand the modified BFS algorithm on the graph below. Enter the values on the "prev" list using the table given.



Vertex	Prev
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	

Problem 2

Below is the label array and the adjacency matrix for a StringGraph.

Part 1 (2 points): Show the "prev" list when BFS is run starting from vertex A.

Part 2 (2 points): What is the shortest path from A to J?

Part 3 (2 points): What is the shortest path from A to M?

Part 4 (2 points): Show the "prev" list when BFS is run starting from vertex Q.

Label Array:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q

Adjacency Matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0		X															
1	X		X	X													
2		X			X	X											
3		X			X												
4			X	X			X	X									
5			X				X										
6					X	X				X							
7					X				X								
8								X		X							
9							X		X								
10												X		X			
11											X		X		X		
12												X					
13											X				X		
14												X		X		X	
15															X		
16																	