

Started on	Monday, 1 April 2024, 7:37 AM
State	Finished
Completed on	Monday, 1 April 2024, 7:42 AM
Time taken	5 mins 20 secs
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

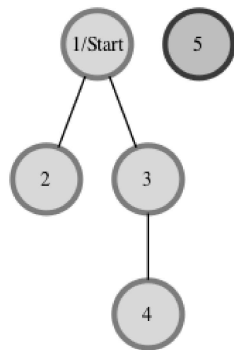
Mark 10.00 out of 10.00

Consider an undirected graph where each edge weighs 6 units. Each of the nodes is labeled consecutively from 1 to n .

You will be given a number of queries. For each query, you will be given a list of edges describing an undirected graph. After you create a representation of the graph, you must determine and report the shortest distance to each of the other nodes from a given starting position using the *breadth-first search* algorithm (BFS). Return an array of distances from the start node in node number order. If a node is unreachable, return -1 for that node.

Example

The following graph is based on the listed inputs:



$n = 5$ // number of nodes

$m = 3$ // number of edges

$edges = [1, 2], [1, 3], [3, 4]$

$s = 1$ // starting node

All distances are from the start node 1. Outputs are calculated for distances to nodes 2 through 5: $[6, 6, 12, -1]$. Each edge is 6 units, and the unreachable node 5 has the required return distance of -1 .

Function Description

Complete the *bfs* function in the editor below. If a node is unreachable, its distance is -1 .

bfs has the following parameter(s):

- *int n*: the number of nodes
- *int m*: the number of edges
- *int edges[m][2]*: start and end nodes for edges
- *int s*: the node to start traversals from

Returns

int[n-1]: the distances to nodes in increasing node number order, not including the start node (-1 if a node is not reachable)

Input Format

The first line contains an integer q , the number of queries. Each of the following q sets of lines has the following format:

- The first line contains two space-separated integers n and m , the number of nodes and edges in the graph.
- Each line i of the m subsequent lines contains two space-separated integers, u and v , that describe an edge between nodes u and v .
- The last line contains a single integer, s , the node number to start from.

Constraints

- $1 \leq q \leq 10$
- $2 \leq n \leq 1000$

- $1 \leq m \leq \frac{n \cdot (n-1)}{2}$
- $1 \leq u, v, s \leq n$

For example:

Input	Result
2 4 2 1 2 1 3 1 3 1 2 3 2	6 6 -1 -1 6
1 5 3 1 2 1 3 3 4 1	6 6 12 -1

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string ltrim(const string &);
6 string rtrim(const string &);
7 vector<string> split(const string &);
8
9 /*
10  * Complete the 'bfs' function below.
11  *
12  * The function is expected to return an INTEGER_ARRAY.
13  * The function accepts following parameters:
14  * 1. INTEGER n
15  * 2. INTEGER m
16  * 3. 2D_INTEGER_ARRAY edges
17  * 4. INTEGER s
18  */
19
20 vector<int> bfs(int n, int m, vector<vector<int>> edges, int s) {
21     vector<vector<int>> newAdjList(n + 1);
22     vector<int> newDistances(n + 1, -1);
23     vector<bool> newVisited(n + 1, false); // Add a visited array to
24     queue<int> newQ;
25
26     for (int i = 0; i < m; i++) {
27         int u = edges[i][0];
28         int v = edges[i][1];
29         newAdjList[u].push_back(v);
30         newAdjList[v].push_back(u);
31     }
32
33     newDistances[s] = 0;
34     newQ.push(s);
35     newVisited[s] = true; // Mark the starting node as visited
36
37     while (!newQ.empty()) {
38         int newNode = newQ.front();
39         newQ.pop();
40
41         for (int newNeighbor : newAdjList[newNode]) {
42             if (!newVisited[newNeighbor]) { // Check if the neighbor
43                 newDistances[newNeighbor] = newDistances[newNode] + 6;
44                 newQ.push(newNeighbor);
45                 newVisited[newNeighbor] = true; // Mark the neighbor
46             }
47         }
48     }
49 }
```

```
47         }
48     }
49
50     vector<int> result;
51     for (int i = 1; i <= n; i++) {
52         if (i != s) {
```

	Input	Expected	Got	
✓	2 4 2 1 2 1 3 1 3 1 2 3 2	6 6 -1 -1 6	6 6 -1 -1 6	✓
✓	1 5 3 1 2 1 3 3 4 1	6 6 12 -1	6 6 12 -1	✓

Passed all tests! ✓

► [Show/hide question author's solution \(C++\).](#)

Correct

Marks for this submission: 10.00/10.00.