All Contests > DAA_LAB > Breadth first search(BFS) 1

Breadth first search(BFS) 1

oblem Submissions Leaderboard Discussions

Write a Java Program to print all nodes reachable from a given starting node in a digraph using BFS method

Input Format

4513022330032

Constraints

positive vertex values only

Output Format

Enter the number of vertices: 4 Enter the number of edges: 5 Enter source: 1 Enter destination: 3 Enter source: 0 Enter destination: 2 Enter source: 2 Enter destination: 3 Enter source: 3 Enter destination: 0 Enter source: 0 Enter destination: 3 Enter Start Vertex for BFS: 2 BFS of graph: 2 3 0

Sample Input 0

Sample Output 0

BFS of graph : 2 3 0

Contest ends in 9 days
Submissions: 117
Max Score: 10
Difficulty: Medium
Rate This Challenge:

f ⊌ in

More



```
3
   import java.text.*;
    import java.math.*;
   import java.util.regex.*;
6 ▼class Gnode {
7
        Gnode next;
8
        int vertex;
 9
   }
10 ▼public class Solution {
        Gnode graph[ ];
11 ▼
        boolean visited[];
12 🔻
        int queue[];
13
        int numVertices;
14
15
        int front;
        int rear;
16
17 ▼
        public Solution(int n) {
            graph = new Gnode[n];
18 🔻
19 🔻
            visited = new boolean[n];
20 🔻
            queue = new int[n];
21
            numVertices = n;
22
            front = -1;
23
            rear = -1;
24
25
        void insertQueue(int vertex) {
26 ▼
            if(rear == numVertices - 1){}
27
            //System.out.printf("Queue Overflow.\n");
28
29
            else {
30
                if(front == -1)
31
                     front = 0;
32
                rear = rear + 1;
33 🔻
                queue[rear] = vertex ;
34
            }
35
        }
36
        boolean isEmptyQueue() {
37 1
38
            if(front == -1 || front > rear)
39
                return true;
40
            else
41
                return false;
        }
42
43
        int deleteQueue() {
44
45
            int deleteItem;
46 ▼
            if(isEmptyQueue()) {
                //System.out.printf("Queue Underflow\n");
47
48
                return -1;
49
            }
            deleteItem = queue[front];
50
51
            front = front + 1;
52
            return deleteItem;
53
        }
        void Bfs(int v) {
54 ▼
            int w;
55
            insertQueue(v);
56
57 1
            while(!isEmptyQueue()) {
58
                v = deleteQueue( );
                System.out.printf(" %d",v);
59
60 •
                visited[v] = true;
                Gnode g = graph[v];
61 ▼
                 for( ; g != null; g = g.next) {
62 ▼
63
                     w = g.vertex;
64 •
                     if(visited[w] == false) {
65
                         insertQueue(w);
66
                         visited[w] = true;
67
                     }
                }
68
            }
69
70
71
        public static void main(String []args) {
            int n, e, i, s, d, v;
72
73
            Gnode q, p;
74
            Scanner sc = new Scanner(System.in);
75
```

```
76
             //System.out.printf("Enter the number of vertices : ");
77
             n = sc.nextInt();
 78
             //System.out.printf("Enter the number of edges : ");
 79
             e = sc.nextInt();
             Solution g = new Solution(n);
80
             for(i=1;i<=e;i++) {
81 1
                 //System.out.printf("Enter source : ");
82
83
                 s = sc.nextInt();
                 //System.out.printf("Enter destination : ");
84
85
                 d = sc.nextInt();
                 q = new Gnode();
86
87
                 q.vertex = d;
88
                 q.next = null;
                 if(g.graph[s] == null)
89 🔻
90 🔻
                     g.graph[s]=q;
91 🔻
                 else {
 92 🔻
                     p=g.graph[s];
93
                     while(p.next != null)
94
                          p = p.next;
95
                      p.next = q;
96
                 }
97
             }
 98
             for(i = 0;i < n;i++)
99 •
                 g.visited[i] = false;
             //System.out.printf("Enter Start Vertex for BFS : ");
100
101
             v = sc.nextInt();
             System.out.printf("BFS of graph :");
102
103
             g.Bfs(v);
104
             System.out.printf("\n");
105
         }
106
    }
107
108
109
110
111
112
113
114
                                                                                                Line: 1 Col: 1
```

<u>♣ Upload Code as File</u> Test against custom input

Run Code

Submit Code

Testcase 0 ✓

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

Your Output (stdout)

```
BFS of graph : 2 3 0
```

Expected Output

BFS of graph : 2 3 0

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |