**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**Department of Computer Application**

**Session: 2023-2024**

**Submission for**

**Course Name:** Design Analysis and Algorithm Lab

**Course Code:** MCP546

**Name of the Student:** Jayesh Lalit Nandanwar

**Class Roll No:** 26

**Semester:** MCA II semester

**Shift:** 2

**Batch:** 2

Under the Guidance of

Prof. Manda Ukey

Date of submission: 10/04/2024

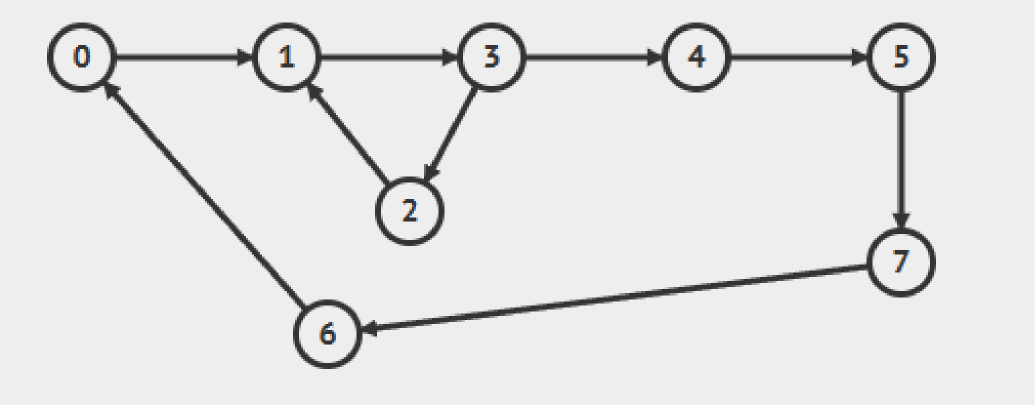
# Practical 8

**Aim:** Perform BFS on a directed graph inputted by the user. The starting node is inputted by the user.
Display the order in which BFS is performed on the graph.
Print the time taken to perform this search. Compare it with DFS.

**Code:**

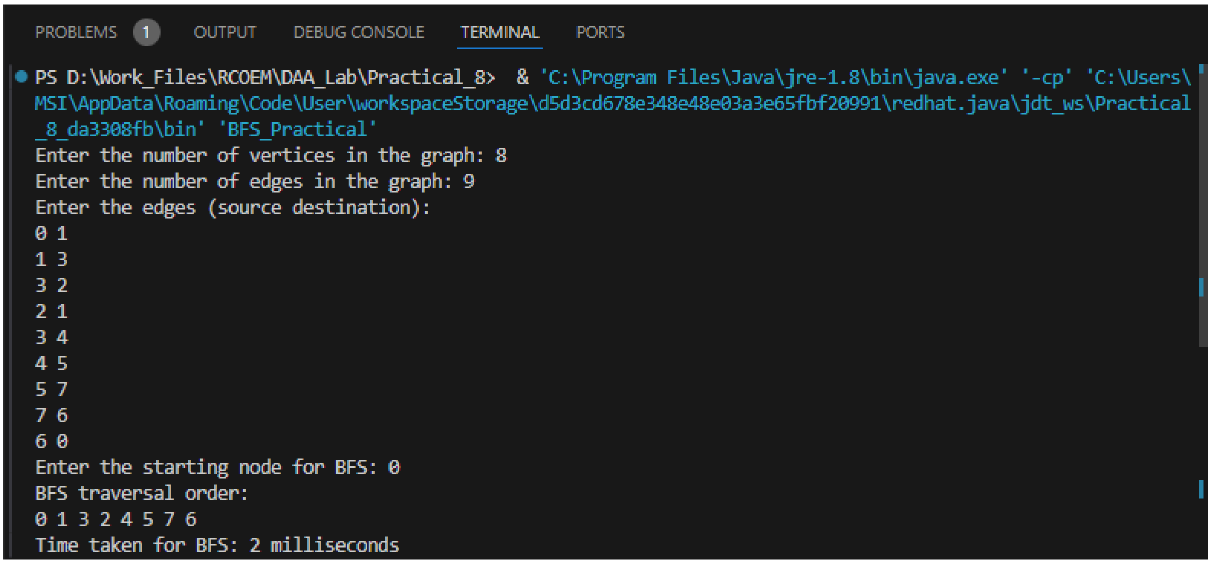
```
import java.util.*;

public class BFS_Practical {
    static class Graph {
        private int V;
        private LinkedList<Integer>[] adj;

        Graph(int v) {
            V = v;
            adj = new LinkedList[v];
            for (int i = 0; i < v; ++i)
                adj[i] = new LinkedList<>();
        }

        void addEdge(int v, int w) {
            adj[v].add(w);
        }

        void BFS(int start) {
            boolean[] visited = new boolean[V];
            LinkedList<Integer> queue = new LinkedList<>();

            visited[start] = true;
            queue.add(start);

            while (queue.size() != 0) {
                start = queue.poll();
                System.out.print(start + " ");

                Iterator<Integer> i = adj[start].listIterator();
                while (i.hasNext()) {
                    int n = i.next();
                    if (!visited[n]) {
                        visited[n] = true;
                        queue.add(n);
                    }
                }
            }
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of vertices in the graph: ");
        int V = scanner.nextInt();
        Graph graph = new Graph(V);

        System.out.print("Enter the number of edges in the graph: ");
        int E = scanner.nextInt();
        System.out.println("Enter the edges (source destination):");
        for (int i = 0; i < E; i++) {
            int src = scanner.nextInt();
            int dest = scanner.nextInt();
            graph.addEdge(src, dest);
        }

        System.out.print("Enter the starting node for BFS: ");
        int startNode = scanner.nextInt();

        long startTime = System.currentTimeMillis();
        System.out.println("BFS traversal order:");
        graph.BFS(startNode);
        long endTime = System.currentTimeMillis();
        long elapsedTime = endTime - startTime;
        System.out.println("\nTime taken for BFS: " + elapsedTime + " milliseconds");

        scanner.close();
    }
}
```

**Graph:**



## Output:

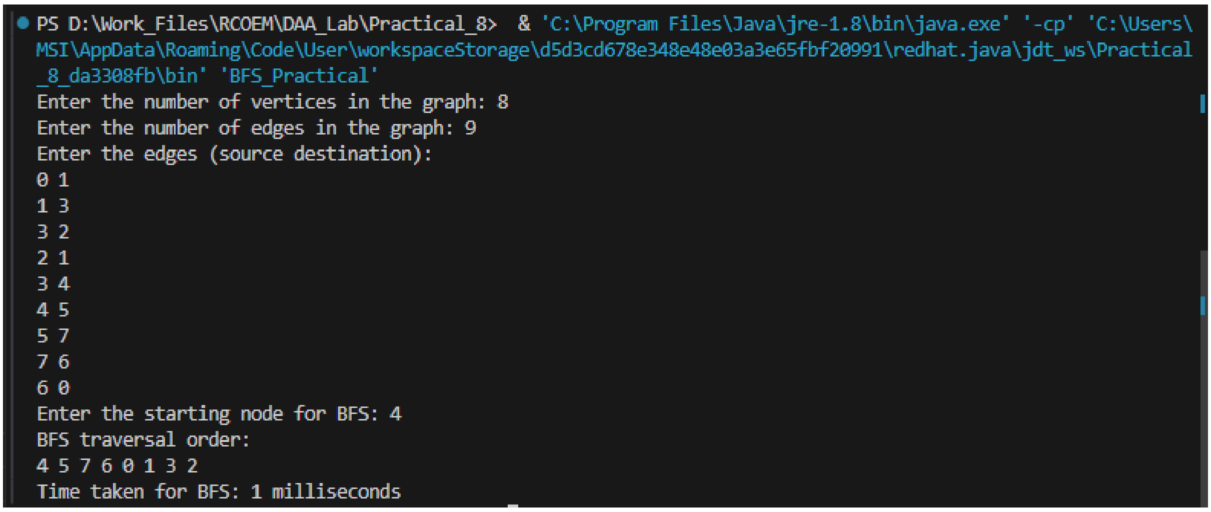
### For Source Node 0:



```
PS D:\Work_Files\RCOEM\DAA_Lab\Practical_8>  & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\
MSI\AppData\Roaming\Code\User\workspaceStorage\d5d3cd678e348e48e03a3e65fbf20991\redhat.java\jdt_ws\Practical
_8_da3308fb\bin' 'BFS_Practical'
Enter the number of vertices in the graph: 8
Enter the number of edges in the graph: 9
Enter the edges (source destination):
0 1
1 3
3 2
2 1
3 4
4 5
5 7
7 6
6 0
Enter the starting node for BFS: 0
BFS traversal order:
0 1 3 2 4 5 7 6
Time taken for BFS: 2 milliseconds
```

**Time Taken: 2 ms**

### For Source Node 4:



```
PS D:\Work_Files\RCOEM\DAA_Lab\Practical_8>  & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\
MSI\AppData\Roaming\Code\User\workspaceStorage\d5d3cd678e348e48e03a3e65fbf20991\redhat.java\jdt_ws\Practical
_8_da3308fb\bin' 'BFS_Practical'
Enter the number of vertices in the graph: 8
Enter the number of edges in the graph: 9
Enter the edges (source destination):
0 1
1 3
3 2
2 1
3 4
4 5
5 7
7 6
6 0
Enter the starting node for BFS: 4
BFS traversal order:
4 5 7 6 0 1 3 2
Time taken for BFS: 1 milliseconds
```

**Time Taken: 1 ms**

**Comparison between time taken:**

| **Searching Algorithm** | **Time Taken (in ms)**<br>**For given graph** | |
|---|---|---|
| | **Source Node: 0** | **Source Node: 4** |
| DFS (Depth First Search) | 1 | 1 |
| BFS (Breadth First Search) | 2 | 1 |