

# **Design Analysis and Algorithm Lab**

**M.C.A.-Semester-I**



**Session: Winter 2024 (2024-2025)**

**Ramdeobaba University, Nagpur**

**Department of Computer Science and  
Application**

**MCA(Artificial Intelligence and Machine  
Learning)**

**Name of Program:** Implementation of Breadth First Search on a Graph and its time complexity with respect to DFS.

**Source Code:**

```
import java.util.*;

public class BFSGraph {

    static int[][] a = new int[20][20];
    static int[] q = new int[20];
    static boolean[] visited = new boolean[20];
    static int n, i, j, f = 0, r = -1;

    static void bfs(int v) {
        q[++r] = v;
        visited[v] = true;

        while (f <= r) {
            int node = q[f++];
            System.out.print(node + "\t");

            // Enqueue all unvisited neighbors of the current node
            for (i = 0; i < n; i++) {
                if (a[node][i] == 1 && !visited[i]) {
                    q[++r] = i;
                    visited[i] = true;
                }
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of vertices: ");
        n = sc.nextInt();

        Arrays.fill(visited, false);
    }
}
```

```

System.out.println("Enter graph data in matrix form (0 for no edge, 1 for edge):");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        a[i][j] = sc.nextInt();
    }
}

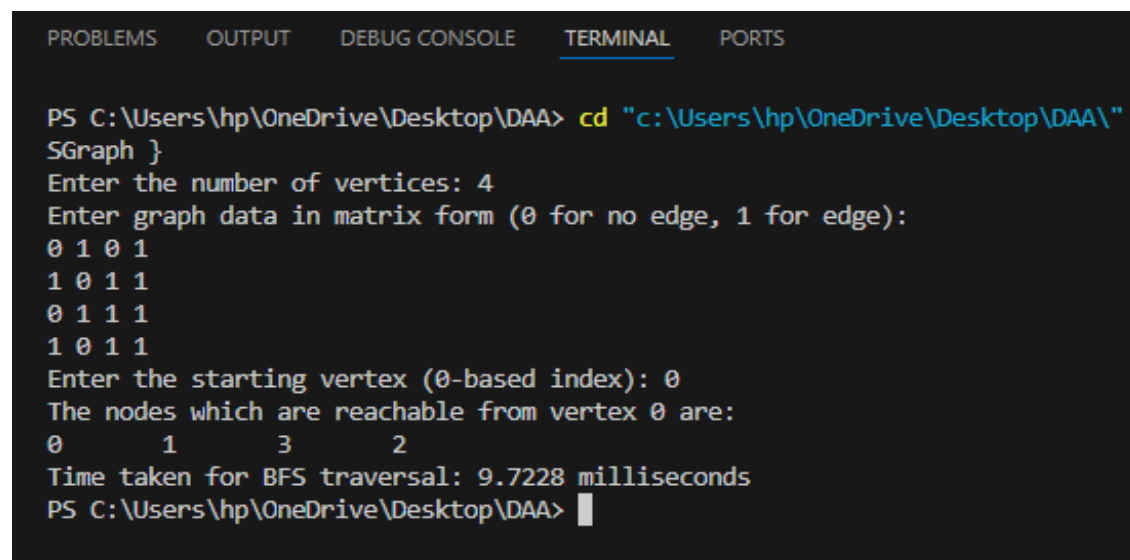
System.out.print("Enter the starting vertex (0-based index): ");
int v = sc.nextInt();
long startTime = System.nanoTime();
System.out.println("The nodes which are reachable from vertex " + v + " are:");
bfs(v);

long endTime = System.nanoTime();
long duration = (endTime - startTime); // Time in nanoseconds
System.out.println("\nTime taken for BFS traversal: " + (duration / 1_000_000.0) + "
milliseconds");

sc.close();
}
}

```

### **Output:**



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\hp\OneDrive\Desktop\DAA> cd "c:\Users\hp\OneDrive\Desktop\DAA\"
SGraph }
Enter the number of vertices: 4
Enter graph data in matrix form (0 for no edge, 1 for edge):
0 1 0 1
1 0 1 1
0 1 1 1
1 0 1 1
Enter the starting vertex (0-based index): 0
The nodes which are reachable from vertex 0 are:
0      1      3      2
Time taken for BFS traversal: 9.7228 milliseconds
PS C:\Users\hp\OneDrive\Desktop\DAA>

```

## **Comparing time complexity with respect to DFS.**

Breadth-First Search (BFS) is another fundamental graph traversal algorithm, but it differs from Depth-First Search (DFS) in how it explores the graph. While DFS explores as deeply as possible down one branch of the graph before backtracking, BFS explores all neighboring nodes level by level before moving to the next level.

**Time Complexity of BFS** is  $O(V + E)$ , just like DFS, but BFS explores the graph level by level and uses a queue, while DFS explores depth-first using a stack (or recursion).

Both algorithms are efficient for graph traversal, and the choice between them depends on the problem requirements, such as whether you need to explore the graph level by level (BFS) or deeply along branches (DFS).

**Submitted By:**

Ritu Choudhary

Roll no:51

MCA(AI/ML) – Section B

