

Assgnment 1

Code:

```
import java.util.*;

class BFS {
    public ArrayList<Integer> bfsOfGraph(int V, ArrayList<ArrayList<Integer>>
adj) {
        ArrayList<Integer> bfs = new ArrayList<>();
        boolean vis[] = new boolean[V];
        Queue<Integer> q = new LinkedList<>();
        q.add(0);
        vis[0] = true;
        while (!q.isEmpty()) {
            Integer node = q.poll();
            bfs.add(node);

            for (Integer it : adj.get(node)) {
                if (vis[it] == false) {
                    vis[it] = true;
                    q.add(it);
                }
            }
        }
        return bfs;
    }

    // DFS logic
    public static void dfs(int node, boolean vis[],
ArrayList<ArrayList<Integer>> adj, ArrayList<Integer> ls) {
        vis[node] = true;
        ls.add(node);
        for (Integer it : adj.get(node)) {
            if (vis[it] == false) {
                dfs(it, vis, adj, ls);
            }
        }
    }

    public ArrayList<Integer> dfsOfGraph(int V, ArrayList<ArrayList<Integer>>
adj) {
        boolean vis[] = new boolean[V + 1];
        vis[0] = true;
        ArrayList<Integer> ls = new ArrayList<>();
        dfs(0, vis, adj, ls);
        return ls;
    }

    public static void main(String args[]) {
        ArrayList<ArrayList<Integer>> adj = new ArrayList<>();
        for (int i = 0; i < 6; i++) {
            adj.add(new ArrayList<>());
        }
        adj.get(0).add(1);
        adj.get(1).add(0);
        adj.get(0).add(4);
        adj.get(4).add(0);
    }
}
```

```

adj.get(1).add(2);
adj.get(2).add(1);
adj.get(1).add(3);
adj.get(3).add(1);
adj.get(3).add(4);
adj.get(4).add(3);
adj.get(4).add(5);
adj.get(5).add(4);

BFS s1 = new BFS();
ArrayList<Integer> ans = s1.bfsOfGraph(6, adj);
int n = ans.size();

System.out.println("Given Graph structure is : ");
for (int i = 0; i < n; i++) {
    System.out.println(" " + i + " -> " + adj.get(i));
}

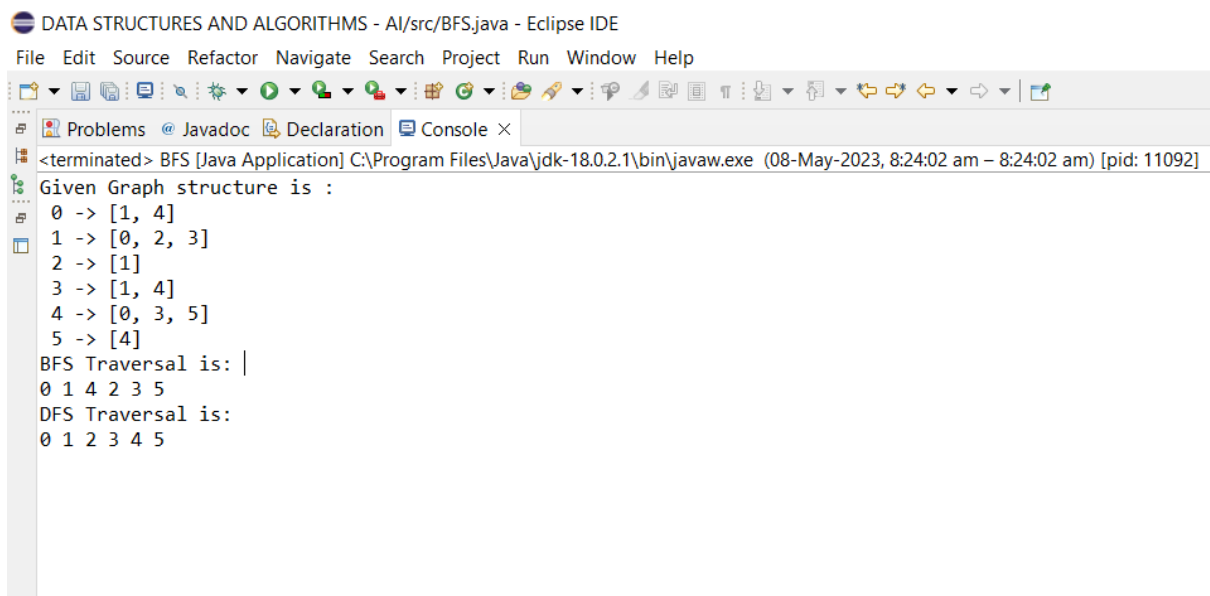
System.out.println("BFS Traversal is: ");
for (int i = 0; i < n; i++) {
    System.out.print(ans.get(i) + " ");
}

ArrayList<Integer> ans2 = s1.dfsOfGraph(5, adj);
int n2 = ans2.size();

System.out.println("\nDFS Traversal is: ");
for (int i = 0; i < n2; i++) {
    System.out.print(ans2.get(i) + " ");
}
}
}

```

Output:



DATA STRUCTURES AND ALGORITHMS - AI/src/BFS.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Problems Javadoc Declaration Console

<terminated> BFS [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (08-May-2023, 8:24:02 am – 8:24:02 am) [pid: 11092]

```

Given Graph structure is :
0 -> [1, 4]
1 -> [0, 2, 3]
2 -> [1]
3 -> [1, 4]
4 -> [0, 3, 5]
5 -> [4]
BFS Traversal is: |
0 1 4 2 3 5
DFS Traversal is:
0 1 2 3 4 5

```