## Green University of Bangladesh

## Department of Computer Science and Engineering

Faculty of Science and Engineering

Semester : (Fall,Year:2021),B.Sc. in CSE(Day)

# Lab Report No.01

**Course title:** Algorithm Lab

**Course Code:** CSE 205 **Section:** D

**Submitted by:**

Sara Shahrin Moumi

Id: 202902023

| | |
|---|---|
| **Course Teacher's Name** | **:** Dr.Faiz Al Faisal |
| **Designation** | **:** Asst. Prof., Dept. of CSE |
| **Submission Date** | **:**12/11/2021 |

```
package topological_sortbfs;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Queue;
import java.util.Scanner;
import java.util.Vector;
class grph {
    int vrt;
    List<Integer> adj[];
    public grph(int vrt)
    {
        this.vrt = vrt;
        adj = new ArrayList[vrt];
        for (int p= 0; p < vrt; p++)
            adj[p] = new ArrayList<Integer>();
    }
    public void add(int u, int v)
    {
        adj[u].add(v);
    }
    public void BfsTopologicalSort()
    {
        int Indegree[] = new int[vrt];
        for (int p = 0; p < vrt; p++) {
            ArrayList<Integer> temp
                = (ArrayList<Integer>)adj[p];
            for (int node : temp) {
```

```
        Indegree[node]++;
    }
}
Queue<Integer> q
    = new LinkedList<Integer>();
for (int p = 0; p < vrt; p++) {
    if (Indegree[p] == 0)
        q.add(p);
}
int c = 0;
Vector<Integer> TopOrder = new Vector<Integer>();
while (!q.isEmpty()) {
    int u = q.poll();
    TopOrder.add(u);

    for (int node : adj[u]) {
        if (--Indegree[node] == 0)
            q.add(node);
    }
    c++;
}
if (c != vrt) {
    System.out.println(
        "cycle iS PRESENT IN the graph");
    return;
}
for (int p : TopOrder) {
    System.out.print(p + " ");
}
```

```
        }
}

public class Topological_sortBFS {

    public static void main(String[] args) {
       grph gp = new grph(14);
                gp.add(12,9);
                gp.add(10,13);
                gp.add(9,11);
                gp.add(9,10);
                gp.add(8,7);
                gp.add(6,5);
                gp.add(5,12);
                gp.add(5,8);
                gp.add(4,7);
                gp.add(3,13);
                gp.add(3,6);
                gp.add(3,2);
                gp.add(2,6);
                gp.add(2,9);
                gp.add(2,5);
                gp.add(1,8);
                gp.add(1,4);
                gp.add(1,2);
                gp.add(0,11);
                gp.add(0,4);
                gp.add(0,5);
```

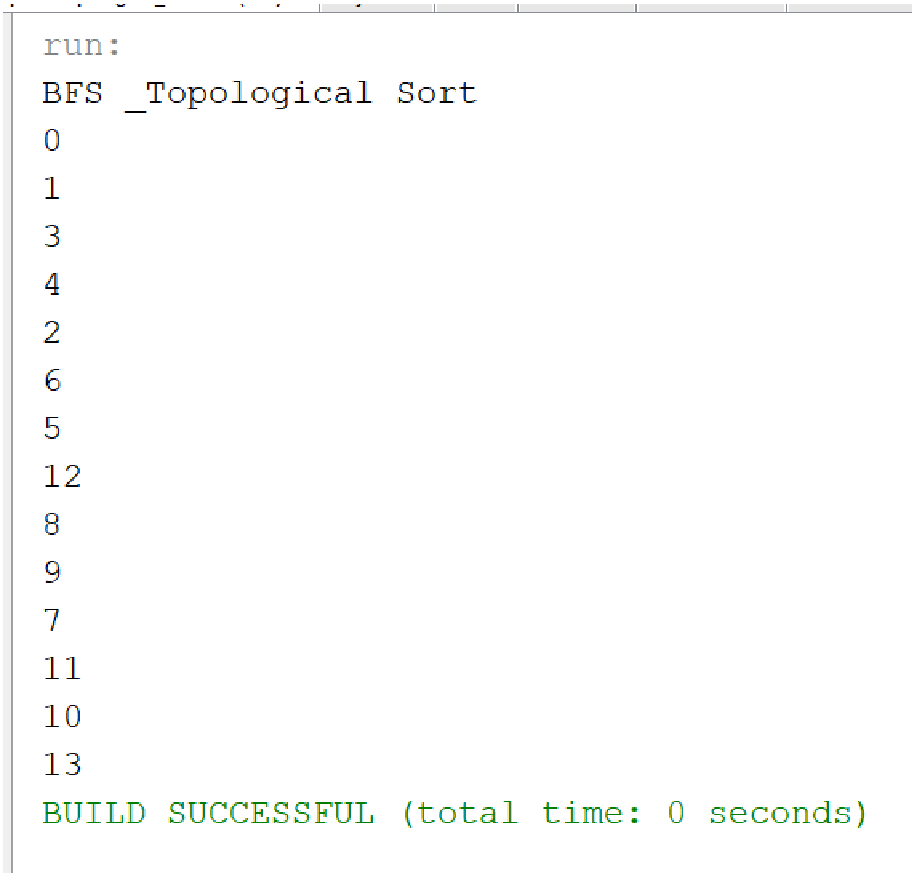
```
            System.out.println("BFS _Topological Sort");

            gp.BfsTopologicalSort();

    }


}
```

OUTPUT:



```
run:
BFS _Topological Sort
0
1
3
4
2
6
5
12
8
9
7
11
10
13
BUILD SUCCESSFUL (total time: 0 seconds)
```