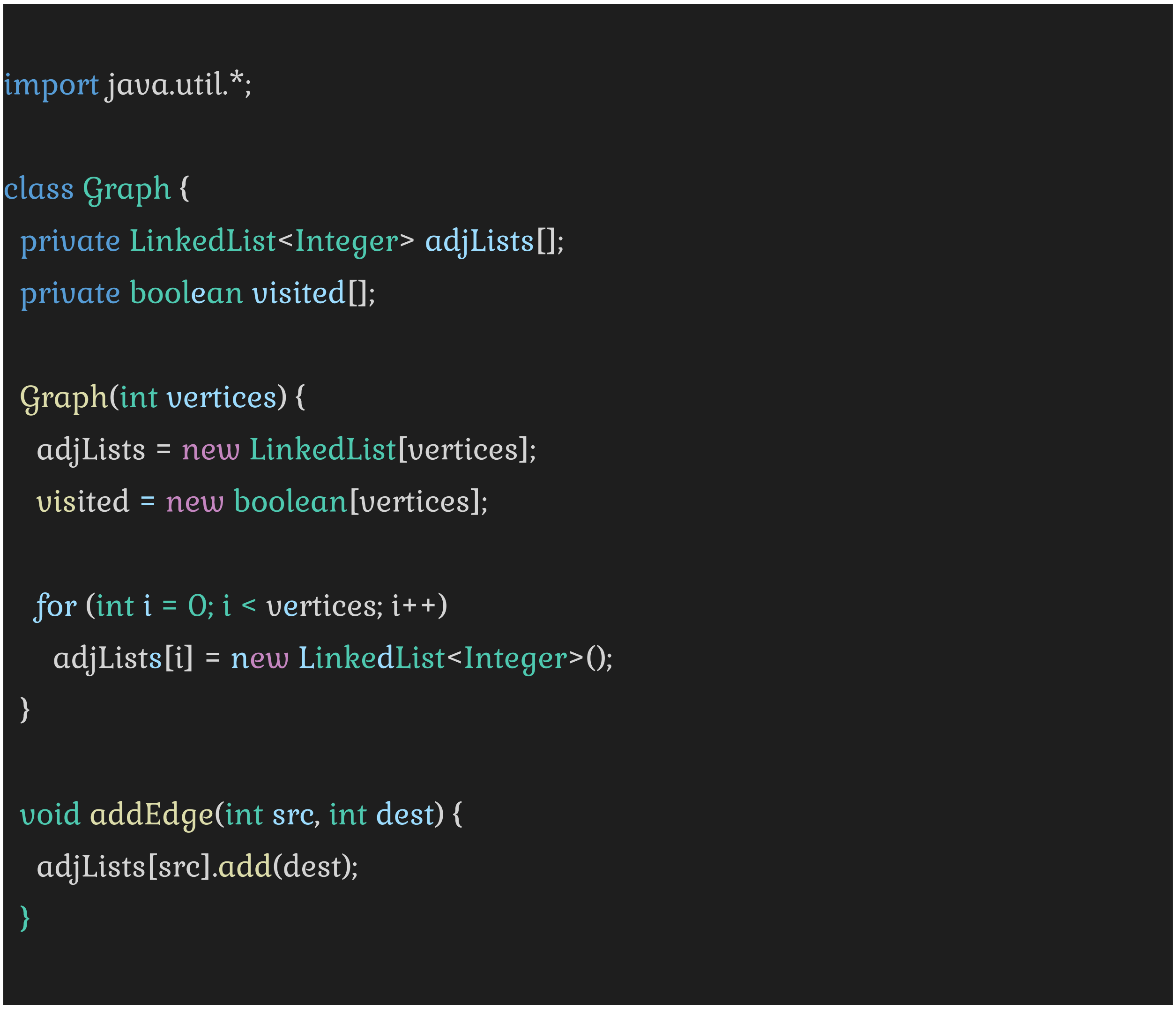
Data Structures and Algorithms

CSE2001

Lab - 6 - Assignment - 2

Yashwanth Reddy
19BCE7362
Date- 16thJuly2021

Problem : Breadth First Search

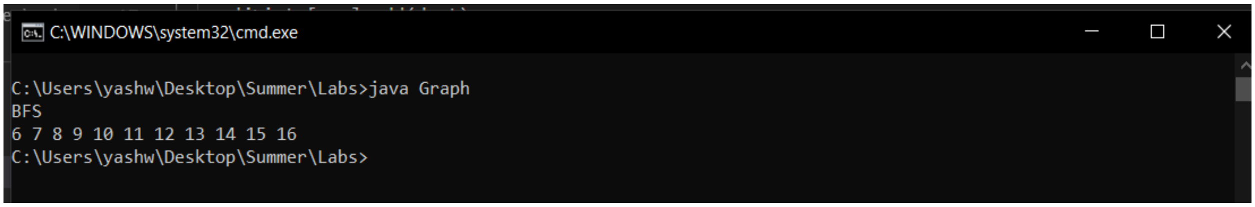


```
import java.util.*;

class Graph {
 private LinkedList<Integer> adjLists[];
 private boolean visited[];

 Graph(int vertices) {
  adjLists = new LinkedList[vertices];
  visited = new boolean[vertices];

  for (int i = 0; i < vertices; i++)
   adjLists[i] = new LinkedList<Integer>();
 }

 void addEdge(int src, int dest) {
  adjLists[src].add(dest);
 }
```

```
void BFS(int s) {



  LinkedList<Integer> queue = new LinkedList();

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

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

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

public static void main(String args[]) {
  Graph g = new Graph(17);

  g.addEdge(0, 1);
  g.addEdge(1, 2);
  g.addEdge(2, 3);
  g.addEdge(3, 4);
```

```
    g.addEdge(4, 5);
    g.addEdge(5, 6);
    g.addEdge(6, 7);
    g.addEdge(7, 8);
    g.addEdge(8, 9);
    g.addEdge(9, 10);
    g.addEdge(10, 11);
    g.addEdge(11, 12);
    g.addEdge(12, 13);
    g.addEdge(13, 14);
    g.addEdge(14, 15);
    g.addEdge(15, 16);



    System.out.println("BFS");
    g.BFS(6);
  }
}
```

Output



```
C:\Users\yashw\Desktop\Summer\Labs>java Graph
BFS
6 7 8 9 10 11 12 13 14 15 16
C:\Users\yashw\Desktop\Summer\Labs>
```