CSC 501 Assignment 2 Programming Problem 6

Name – Jaspreet Singh
Student ID – 210555347
Email ID – fjaspreetsingh1@toromail.csudh.edu

Requirements:

1. **Accepting a positive integer n from keyboard and then creating a random undirected graph with n nodes.**

```
public static void main(String args[])
{
	int count  = 1, source, dest;

	System.out.print("Enter the number of vertices");
	Scanner scan = new Scanner(System.in);
	number_vertices = scan.nextInt();
	int number_edges  = ThreadLocalRandom.current().nextInt(0, (number_vertices-
1)*(number_vertices-1));
	System.out.println("Edges are: "+number_edges);
	GraphAdjacencyList adjacencyList = new GraphAdjacencyList(number_vertices);
	while(count <= number_edges)
	{
		source = ThreadLocalRandom.current().nextInt(1, number_vertices);
		dest = ThreadLocalRandom.current().nextInt(1, number_vertices);
		if(source != dest)  // Does't create edge for self loop
		{
			adjacencyList.addEdges(source, dest);
		}
		count++;
	}
	System.out.println("The given adjacency List for the graph\n");
	for(int i=1; i<number_vertices; i++)
	{
		System.out.print(i+"->");
		edgeList = adjacencyList.getNeighbours(i);

		if(edgeList.size() > 0)
		{
			for(int j=1;;j++)
			{
				if(j!=edgeList.size())
				{
					System.out.print(edgeList.get(j-1)+" -> ");
				}
				else
				{
					System.out.println(edgeList.get(j-1));
					break;
```

```
                        }
                    }
                    System.out.println();
                }

        }
public GraphAdjacencyList(int vertices)
{
        adjListsMap = new HashMap<Integer, HashSet<Integer>>();
        for(int i =1; i<=vertices;i++)
        {
                HashSet<Integer> neighbours = new HashSet<Integer>();
                adjListsMap.put(i, neighbours);
        }
}
public void addEdges(int u, int v)
{
        if(u > adjListsMap.size() || v > adjListsMap.size())
        {
                return;
        }
         (adjListsMap.get(u)).add(v);
}
public ArrayList<Integer> getNeighbours(int u)
{
        if(u > adjListsMap.size())
        {
                return null;
        }
        return new ArrayList<Integer>(adjListsMap.get(u));
}
```

2. **No self-looping edge for each node:**

```
if(source != dest)  // Does't create edge for self loop
{
        adjacencyList.addEdges(source, dest);
}
```

Execution: Output Screenshots

1. With the number of vertices 8

```
C:\Users\Jaspreet Singh\Desktop\CSC-501>javac GraphAdjacencyList.java

C:\Users\Jaspreet Singh\Desktop\CSC-501>java GraphAdjacencyList
Enter the number of vertices8
Edges are: 48
The given adjacency List for the graph

1->2 -> 3 -> 5 -> 6 -> 7

2->1 -> 3 -> 4 -> 5 -> 6 -> 7

3->1 -> 2 -> 4 -> 5 -> 6 -> 7

4->2 -> 3 -> 5 -> 7

5->1 -> 2 -> 3 -> 4 -> 6 -> 7

6->1 -> 2 -> 3 -> 5 -> 7

7->1 -> 2 -> 3 -> 4 -> 5 -> 6

Please enter the start node for the BFS between 1 and 8
4
BFS Tree:
 4
 2 (4) 3 (4) 5 (4) 7 (4) 1 (2) 6 (2)
C:\Users\Jaspreet Singh\Desktop\CSC-501>
```

2. With number of vertices 10

```
C:\Users\Jaspreet Singh\Desktop\CSC-501>javac GraphAdjacencyList.java

C:\Users\Jaspreet Singh\Desktop\CSC-501>java GraphAdjacencyList
Enter the number of vertices10
Edges are: 17
The given adjacency List for the graph

1->3 -> 5

2->3 -> 5 -> 6 -> 7 -> 8

3->1 -> 2 -> 5

4->5->1 -> 2 -> 3 -> 7

6->2 -> 7 -> 9

7->2 -> 5 -> 6 -> 9

8->2

9->6 -> 7

Please enter the start node for the BFS between 1 and 10
9
BFS Tree:
 9
 6 (9) 7 (9) 2 (6) 5 (2) 3 (2) 8 (2) 1 (3)
C:\Users\Jaspreet Singh\Desktop\CSC-501>
```

3. With the number of vertices 5

```
C:\Users\Jaspreet Singh\Desktop\CSC-501>javac GraphAdjacencyList.java

C:\Users\Jaspreet Singh\Desktop\CSC-501>java GraphAdjacencyList
Enter the number of vertices5
Edges are: 9
The given adjacency List for the graph

1->2 -> 4

2->1 -> 3 -> 4

3->2

4->1 -> 2

Please enter the start node for the BFS between 1 and 5
3
BFS Tree:
 3
 2 (3) 1 (2) 4 (2)
```

4. With the number of vertices 20

```
C:\Users\Jaspreet Singh\Desktop\CSC-501>javac GraphAdjacencyList.java

C:\Users\Jaspreet Singh\Desktop\CSC-501>java GraphAdjacencyList
Enter the number of vertices20
Edges are: 120
The given adjacency List for the graph

1->16 -> 18 -> 3 -> 4 -> 5 -> 8 -> 13 -> 14

2->16 -> 19 -> 4 -> 5 -> 6 -> 7 -> 14

3->1 -> 18 -> 19 -> 5 -> 7 -> 8 -> 11 -> 12 -> 13 -> 14 -> 15

4->17 -> 1 -> 2 -> 18 -> 19 -> 7 -> 9 -> 10 -> 12 -> 13 -> 15

5->1 -> 2 -> 18 -> 19 -> 3 -> 9 -> 13 -> 14 -> 15

6->16 -> 17 -> 2 -> 18 -> 19 -> 8 -> 9 -> 10 -> 11 -> 14

7->16 -> 2 -> 3 -> 4 -> 9 -> 10 -> 11 -> 13 -> 14 -> 15

8->1 -> 17 -> 18 -> 3 -> 6 -> 11 -> 13 -> 14

9->18 -> 19 -> 4 -> 5 -> 6 -> 7 -> 10 -> 11 -> 13 -> 14 -> 15

10->18 -> 4 -> 6 -> 7 -> 9 -> 12 -> 13

11->17 -> 18 -> 3 -> 6 -> 7 -> 8 -> 9 -> 13 -> 14
```

```
12->17 -> 3 -> 4 -> 10 -> 14 -> 15

13->17 -> 1 -> 19 -> 3 -> 4 -> 5 -> 7 -> 8 -> 9 -> 10 -> 11 -> 15

14->1 -> 2 -> 3 -> 5 -> 6 -> 7 -> 8 -> 9 -> 11 -> 12 -> 15 -> 16 -> 18 -> 19

15->16 -> 17 -> 3 -> 19 -> 4 -> 5 -> 7 -> 9 -> 12 -> 13 -> 14

16->1 -> 2 -> 6 -> 7 -> 14 -> 15

17->19 -> 4 -> 6 -> 8 -> 11 -> 12 -> 13 -> 15

18->1 -> 3 -> 4 -> 5 -> 6 -> 8 -> 9 -> 10 -> 11 -> 14

19->17 -> 2 -> 3 -> 4 -> 5 -> 6 -> 9 -> 13 -> 14 -> 15

Please enter the start node for the BFS between 1 and 20
8
BFS Tree:
 8
 1 (8) 17 (8) 18 (8) 3 (8) 6 (8) 11 (8) 13 (8) 14 (8) 16 (1) 4 (1) 5 (1) 19 (3) 12 (3) 15 (3) 9 (18) 10 (18) 7 (3) 2 (19
)
C:\Users\Jaspreet Singh\Desktop\CSC-501>
```