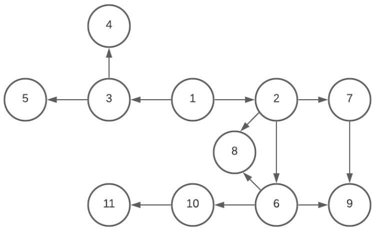
Yeditepe University
CSE 211 - Data Structures
Assignment 3

Deadline: 5 December 2021, 23:59

In this assignment, you are given a text file that represents the direct motorways from city ‘1’ to the other 10 cities in *The Shire* which is a fictional region. You will implement a directed graph class via adjacency lists to perform some algorithms. You must use linked lists while implementing adjacency lists. The Graph should have a dynamic size. You must use the queue and the link list structures provided in the assignment files.



Below are constructors and functions that you need to implement in this assignment:

```
intDrGraph(string filename);
```

This constructor will build the adjacency list from the contents of a file that will be taken from the user as a parameter. The file includes a single number in the first line which shows the number of vertices in the graph. Then a number of lines consist of two integers, one of them is the source vertex and the other one is the destination vertex for an edge.

For the example graph given, the contents of the file:

```
11
1 2
1 3
3 4
3 5
2 6
2 7
2 8
7 9
6 9
6 8
6 10
10 11
```

```
void printAdjList();
```

- This function should print the adjacency list.

```
void performBFS(int start_index);
```

- This function should perform the Breadth-First-Search traversal starting from the start_index provided in the parameter.

```
bool isThereWay (int src_city, int dest_city);
```

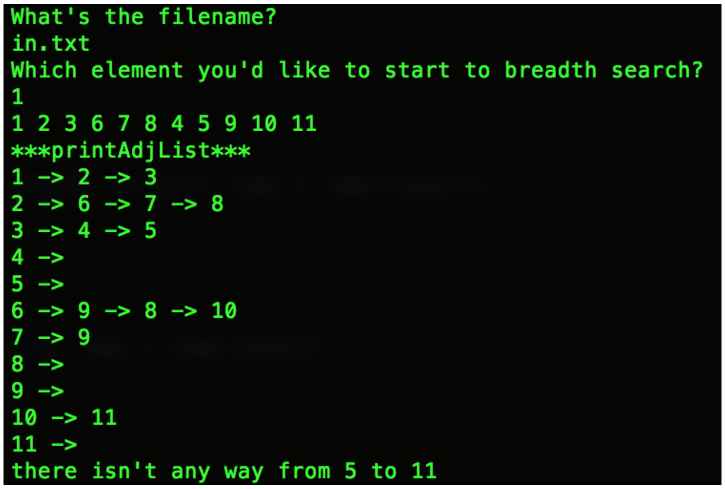
- This function will find out whether there is a way to go **from** src city **to** dest city vertices. Of course, there is a way from city 1 to all other cities. However, for example, there is no way to go from 5 to 11.

BONUS:

```
void findRoutes (int src_city, int dest_city);
```

- This function will find out and print the possible paths from src_city to dest_city. For example, from 1 to 8, possible paths are:
    - 1 2 8
    - 1 2 6 8

Below figure shows the execution snippet for the graph given:



```
What's the filename?
in.txt
Which element you'd like to start to breadth search?
1
1 2 3 6 7 8 4 5 9 10 11
***printAdjList***
1 -> 2 -> 3
2 -> 6 -> 7 -> 8
3 -> 4 -> 5
4 ->
5 ->
6 -> 9 -> 8 -> 10
7 -> 9
8 ->
9 ->
10 -> 11
11 ->
there isn't any way from 5 to 11
```

Do your own work.
.zip your files and upload the .zip file in the form of *NAME_SURNAME_ID.zip*