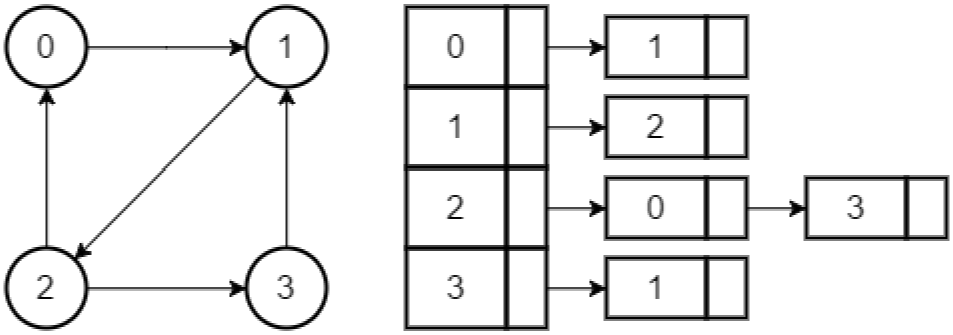
DSA/CS 5005 Computing Structures – Programming Project – BONUS – Fall 2016
(Due: Dec 9th, 2016 11:59 PM)

Objectives

1. [10 pts] Correctly implement the graph data structure
2. [10 pts] Implement Depth First Search
3. [10 pts] Implement Breadth First Search
4. [20 pts] Implement main() to process the commands correctly
5. Document your project thoroughly as the examples in the textbook. This includes, but is not limited to, header comments for all classes/methods, explanatory comments for each section of code, meaningful variable and method names, and consistent indentation. Program that lack or are weak in documentation will receive a deduction of up to 30% of the grade

Project Description

In this project, you will be implementing a directed graph data structure



Above is a simple directed graph with its corresponding adjacency list. You will be implementing a graph data structure using an adjacency list.

```
class Graph
{
private:
        int numOfNodes; // Number of nodes in the graph
        list<int> *adjList; // The adjacency list
        // other private methods
public:
        // public methods
};
```

The adjacency list will be represented by an array of lists, where the index of the array holds the adjacency list for that node. For example, if the example graph was implemented,

adjList[0] = (1) and adjList[2] = (0, 3)

Methods that should be implemented:

- void addEdge(int v, int w) **–** Adds the edge from node v to node w. If the edge already exists, this method should alert the user and skip adding the edge.
- void removeEdge(int v, int w) – Removes the edge from node v to node w. If the edge doesn't exist, this method should alert the user.
- void listNeighbors(int n) – Should print out all neighbors of the selected node.
  For Example: "Neighbors of node 2 are: 0, 3"
- void DFS(int n) – Preforms Depth First Search with n being the starting node.
  Output should look like "DFS starting at node 2: 0, 1, 3"
- void BFS(int n) – Preforms Breadth First Search with n being the starting node.
  Output should look like "BFS starting at node 2: 0, 3, 1"

Input File

The input file will be like the input files from project 1 and 2

```
5

ADD 0 1

ADD 1 2

ADD 2 3

ADD 3 1

ADD 2 0

LIST 2

DFS 2

BFS 2

REMOVE 1 2
```

The first line will be the number of nodes in the graph

ADD lines should call the addEdge method with the two numbers as arguments

LIST lines should call the listNeighbors method with the number as its argument

DFS lines should call the DFS method with the number as its argument

BFS lines should call the BFS method with the number as its argument

REMOVE lines should call the removeEdge method with the number as its argument

Constraints

1. This project must be implemented in C++. The only header file you will have <iostream>, <list> and <cstdlib> (for atoi() which can be used to convert a C string to an integer in the input file). We will not use any other libraries.
2. None of the projects will be a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.