

# CS 2413: Data Structures – Programming Project 5 – Spring 2019

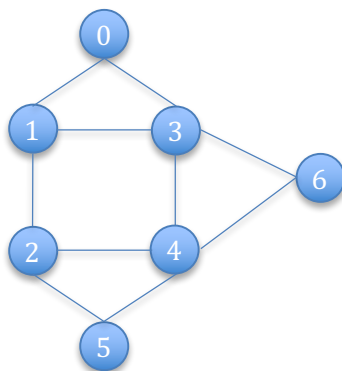
## Due 11:59 PM, April 28, 2019

### Objectives

1. [30 pts] Graph Class for each representation: adjacency matrix and adjacency list
2. [10 pts] Add Edge
3. [10 pts] Using Queue Standard Template Library for use as part of BFS (see below)
4. [10 pts] Ostream operator to print the graph in the edge list format
5. [10 pts] Implement the copy constructor and the overloaded = (assignment) operator
6. [30 pts] Breadth First Search (BFS) and output the parent array representation of the BFS tree. This function will take as input the Graph object and the start vertex p.
7. [20 pts Bonus] Perform Depth First Search (DFS) and output the parent array representation of the DFS tree. This function will take as input the Graph object and the start vertex p.
8. Document your project thoroughly as the examples in the textbook. This includes but 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 weak in documentation will receive a *deduction* of up to 30% of the grade.

### Project Description

A graph  $G = (V, E)$  consists of vertex set  $V$  and the edge set  $E$ . We will assume the graph  $G$  is undirected implying that the edges have no direction. We will also assume that the graph is connected (there exists a path between every pair of nodes).



7	Number of nodes
2 4	Edge from 2 to 4
0 1	
3 0	
3 6	
6 4	
3 4	
1 3	
1 2	
5 2	
4 5	

### Input File

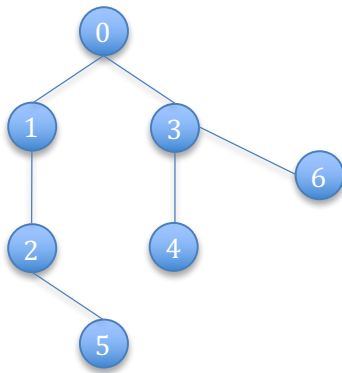
The first line of the input contains an integer that corresponds to the number of nodes in the input graph. Followed by this integer, there will be unknown number of lines of input. Each line of input will contain two integer  $u$  and  $v$ , indicating that there is an edge between vertices  $u$  and  $v$ .

### Processing and Output

In the main program create an object of AdjacencyMatrixGraph and AdjacencyListGraph after reading the number of vertices and calling the appropriate constructor. Next, after reading each edge  $u$  and  $v$ , call the addEdge method on the graph.

After the graph is created you need to write C++ statements to demonstrate copy constructor and overloaded = operator.

After the graph is created you will call a function that will perform the Breadth First Search. This algorithm (and C++ method) is provided in your textbook. The output of this function will be a parent array to store the parent of each node in the Breadth First Search. If we assume that vertex 0 is the start vertex, the BFS search tree and the parent representation is shown below.



0 1 2 3 4 5 6 <- Nodes  
-1 0 1 0 3 2 3 <- Parent Number

### Constraints

1. This project must be implemented in C++. The only header file you will have <iostream>, <list> and <queue> standard template 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.