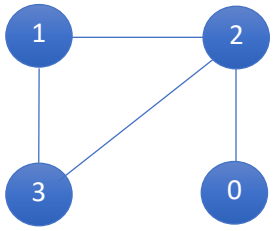
CS/DSA 5005 – Computing Structures – Project 3 – Summer 2019
Due August 15th, 2019, 11:59 PM

Objective: In this project, you will learn and implement Breadth First Search(BFS) and Depth First Search(DFS) using the Compressed Sparse Row Data Structure which we had already used in our first project and store the two trees(BFS and DFS) in the parentArray data structure from our second 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).



4 4 5 ← n, m and numNZV
0 0 1 0 ← the adjacency matrix
0 0 1 1
1 1 0 1
0 1 1 0

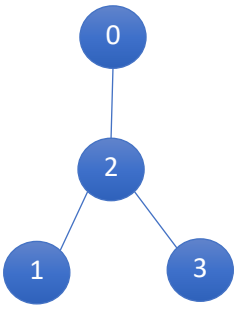
0 ← start node for BFS and DFS

Input file: The input file will be similar to the one from the first project. Only one input matrix is given in the input file and an example is shown above. You will have a the number of rows(n), number of columns(m) and the number of non-zero values(numNZV) in the first line of the input file. Following will be a boolean matrix of dimension n x m. This is the adjacency matrix with which can reconstruct a graph. The last line after the matrix is the node with which you have to start with for your BFS and DFS traversal.

Explanation: The Adjacency matrix is a way to represent a graph structure. The number of rows and number of columns are the always equal and are the number of nodes present in the graph. If there is a 1 present in a spot, that means there is an edge going from the node number in the row to the node number in the column. The graph is assumed to be non-directional and also not have a weight for its edges.

The goal is to perform BFS and DFS on the given graph from the start node and store the resulting tree in the parentArray data structure which we have done through in the second project.

The BFS tree of the above graph will be:



The parentArray for this BFS tree starting with the node 0 which is given will be the following:

0	1	2	3
-1	2	0	2

A sample input file has been given. A sample output file will be given as well. The project submissions should be done in GradeScope where it will be auto graded.

Constraints:

- In this project, the only header you will use is #include <iostream> and the **#include<queue>** stl library for using queues for the BFS implementation.
- None of the projects is 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.