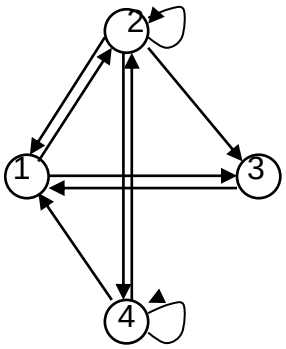
605.202 Data Structures and Algorithms Lab 2

The diagram below is an example of a directed graph. Each edge has an arrow denoting its direction. Node B is considered adjacent to node A if there is a directed edge from A to B. A will only be adjacent to B if there is also an edge from B to A. It is possible to associate a matrix called the adjacency matrix with a graph. Adj[i,j] = 1 if and only if there is an edge from node i to node j in the graph. If Adj[i,j] = 0 then there is no edge from node i to node j. (You may find it helpful to review the chapter in the text on graphs). We will assume that our adjacency matrices contain only 1's or 0's.

In a graph, one is usually interested in the possible paths from one node to another. The ones which are the most useful are the ones which do not contain loops (cycles), i.e.. do not visit a node more than once. The node sequence 1 2 4 is an example of a path with no loops. 1 2 2 4 1 3 is an example of a with a cycle and a loop. An allowable exception is for the starting and ending node to be the same. 1 2 1 is OK but not 1 2 1 2 4 1.



"To Nodes"

	0	1	1	0	Adjacency
"From	1	1	1	1	Matrix
Nodes"	1	0	0	0	
	1	1	0	1	

Write a program to read in the number of nodes in the graph and the corresponding adjacency matrix, one row at a time. Echo the adjacency matrix to your output file. The program should use recursion to find and list all possible non-looping paths between all possible pairs of nodes. It is ok if the start and end nodes are the same but otherwise the nodes should not repeat within the path. If no path exists for a particular pair of nodes then print 'No Path Found'. Check out all possible paths in each graph. See the attached file for the required input. **Make up some other graphs as additional input.** You will lose points for having insufficient I/O.

The recursion is the most important part of this assignment. Use an array to store the adjacency matrix. Consider why it is appropriate. You may not use ArrayLists. Consider an iterative solution as part of your analysis. What impact would that have on your data strucutres.

Remember, you are responsible for showing that your program does all that it is supposed to do. If you put in any special features, you need to have an I/O set which demonstrates the features

Your program should read an adjacency matrix from the file and process the graph to find all the paths, then read an new adjacency matrix, until the file is empty. Make sure to allow for reasonable error situations.