Laboratory practice No. 5: Graph Implementation

Pablo Alberto Osorio Marulanda
Universidad Eafit
Medellín, Colombia
paosoriom@eafit.edu.co

Verónica Mendoza Iguarán
Universidad Eafit
Medellín, Colombia
vmendozai@eafit.edu.co

3) Practice for final project defense presentation

1. The implementation of graphs with matrix data structure contains an integer matrix that stores the weight of the node in each position. The size of the matrix is given by the number of nodes in the graph, the rows represent the origin and the columns the destination or vertex to which the node points. The implementation with adjacency lists data structure, consists of a list with as many positions as nodes has the graph; where each position of the list is a node of the graph, and each position has a list with integer pairs: an integer is the destination of the node and the other is its weight.
2. In this case, it is better to use an implementation of a graph using ArrayList because for this case we need to insert so many data in the graph, and the method which have less complexity to insert is a method implemented with ArrayLists.
3. If our goal is saving RAM, the best option is to use Adjacency Lists, as this would have a size of 100 million (number of Facebook users) and each position in the list would have smaller lists (with the number of friends). that each user has). If implemented with matrices, the size would always be 100 million squared. however, the implementation that will be given to the program depends on it because some methods are less efficient.
4. By means of an adjacency matrix, we can know what is the distance between a node and another simply by knowing the origin and destination; whereas with an adjacency list we need to use a cycle to go through the lists; so the representation of a routing table requires matrices because what these tables seek is to know the distances from one device to another and in this way the complexity is less.
5. recorrer 2.5:
   The complexity is O(n*m)
6. 2.5:
   The n variable means the total of pairs in the array parejas*, and* the m variable means the total of nodes in the graph.

4) Practice for midterms

1.

	0	1	2	3	4	5	6	7
0				1	1			
1	1		1			1		
2		1			1		1	
3								1
4			1					
5								
6			1					
7								

2. **0-** [3,4]
1- [0,2,5]
2- [1,4,6]
3- [7]
4- [2]
5- []
6- [2]
7- []

3. *b*

5) Recommended reading (optional)

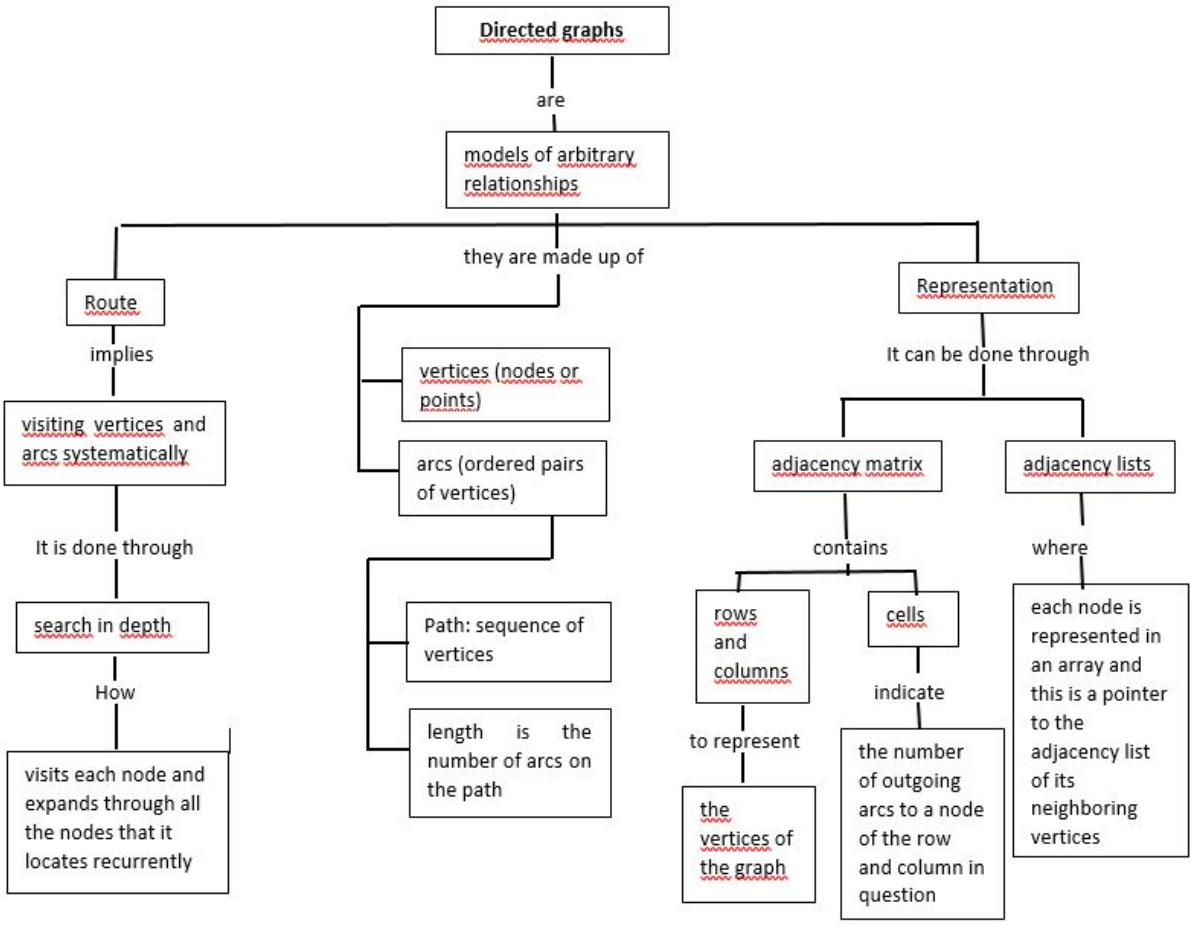
Directed graphs

a) Directed graphs are models of arbitrary relationships, usually used to represent possible short paths. A graph consists of vertices (nodes or points) and arcs (ordered pairs of vertices). In a graph, a path is a sequence of vertices and the length is the number of arcs on the path. Sometimes labeled directed graphs are used, in these, each arc or vertex has a label (name, cost or value).
The representation of a directed graph can be done by means of an adjacency matrix in which the rows and columns are identified with the vertices of the graph and the cells indicate the number of outgoing arcs to a node of the row and column in question; or by adjacency lists, where each node is represented in an array and this is a pointer to the adjacency list of its neighboring vertices.

PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co

The use of graphs is important because they relate algorithms that simplify processes, an example of this are the Dijkstra and Floyd algorithms, which allow finding the shortest path from one vertex to other vertices of the graph.
The route of a directed graph involves visiting vertices and arcs systematically. The search in depth, allows to achieve it in an orderly manner, as it visits each node and expands through all the nodes that it locates recurrently.
Another type of graphs directed are the acyclic ones: these are graphs without cycles (they start and end in different vertex) and are useful for the representation of arithmetic expressions or the representation of partial orders.

b) Concept map



PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co

6) Team work and gradual progress (optional)

Member	Date	Done	Doing	To do
Pablo	17/10/2018 Start time: 2:40pm		· online exercise (assign colors to the nodes without repeating)	· analyze the problem online for your best solution
Verónica	17/10/2018 Start time: 2:40pm	· Implementation of graphs with the matrix data structure and with Lists of Adjacency.		· Explanation of the operation of exercise 1.1
Pablo	19/10/2018 Start time: 1:00pm	· continuation of the online exercise		
Verónica	19/10/2018 Start time: 1:00pm	· Explanation of the operation of exercise 1.1 · Analysis of the question: What is better to use, Adjacency Matrices or Adjacency Lists?		· inquire about advantages and disadvantages of matrices and adjacency lists

Pablo	20/10/2018 Start time: 2:10pm	· Analysis of the question: What is better to use, Adjacency Matrices or Adjacency Lists?	· inquire about advantages and disadvantages of matrices and adjacency lists	
Verónica	20/10/2018 Start time: 2:10pm	· Implementation of method that returns the vertex with the most successors		
Pablo	21/10/2018 Start time: 11:35am	· calculation of the complexity of the online exercise · Explanation of the complexity calculation variables		
Verónica	21/10/2018 Start time: 11:35am	· Compilation of all exercises and improvements		