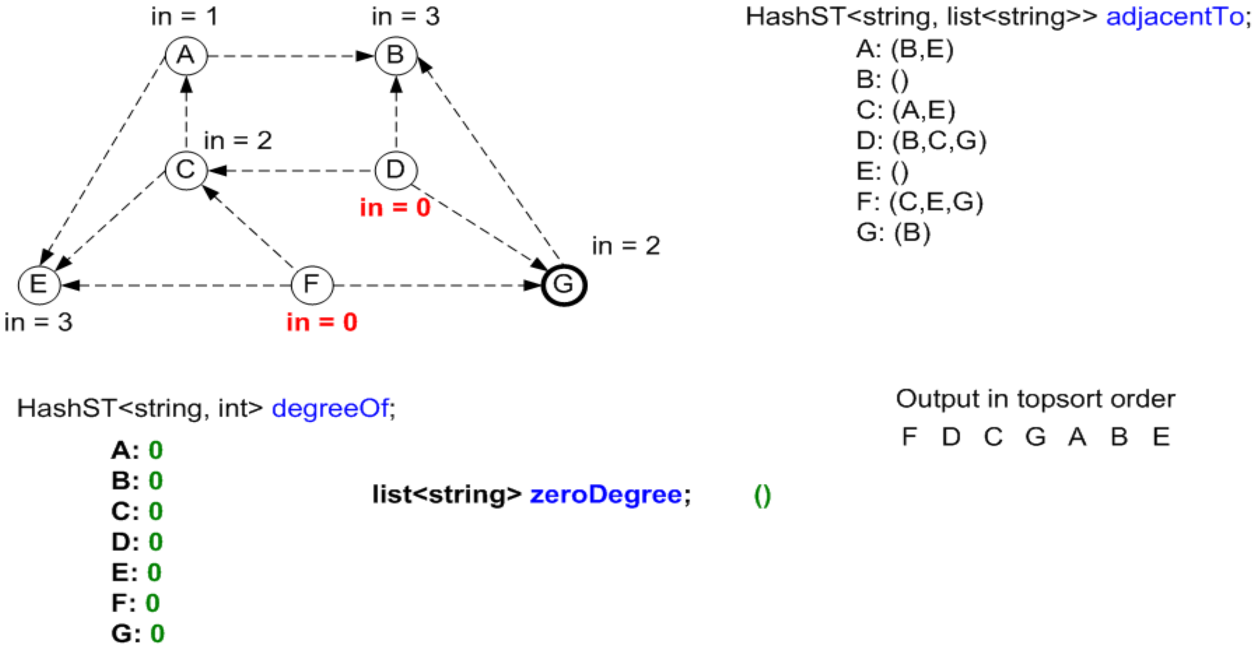
Laboratory practice No. 1: Graph implementation

Juan Sebastián Pérez Salazar
Universidad Eafit
Medellín, Colombia
jsperezs@eafit.edu.co

Yhoan Alejandro Guzmán García
Universidad Eafit
Medellín, Colombia
yaguzmang@eafit.edu.co

3) Practice for final project defense presentation

3.1 For storing the graph we used a HashMap and LinkedLists. We used a HashMap for storing each node because a fast access is needed when processing graphs, and also because the nodes ID's would not provide a proper numerical sequence to use regular arrays. The LinkedLists were used to store each node's neighbors since it is not known how many neighbors each node has. The node's LinkedLists store a pair of a string and an arc. The string contains the ID of the neigthbor, and the arc contains the ID of the node where it comes from and the ID of the node it goes to, as well as the distance and the name.
The next image shows a similar data structure:



```
HashST<string, list<string>> adjacentTo;
    A: (B,E)
    B: ()
    C: (A,E)
    D: (B,C,G)
    E: ()
    F: (C,E,G)
    G: (B)
```

```
HashST<string, int> degreeOf;
    A: 0
    B: 0
    C: 0
    D: 0
    E: 0
    F: 0
    G: 0
```

```
list<string> zeroDegree;      ()
```

```
Output in topsort order
F  D  C  G  A  B  E
```

(taken from: http://condor.depaul.edu/glancast/393class/docs/lecMar14.html**)**

3.2 Assuming that each position in the matrix consumes 1 byte, the whole matrix would consume 90 gigabytes.

3.3 We did not have any problem with the IDs**.**

3.4 2.1The algorithm basically goes throw each node assigning a color to its neighbors, there is an array with the available colors so no color is repeated. When all neighbors are colored, if the amount of colors used was more than 2 then the node is not bicolorable and so is not the graph.

2.3 The name of the algorithm is the claw. This algorithm defines if a graph is bipartite. What is done in this algorithm is that the data of the graph is stored in a HashMap and it is defined if it is bipartite or not by means of some logical conditions. This algorithm can define if the graph is or is not bipartite before finishing reading the sequence of entries, it is also done at a fairly acceptable speed.

3.5 2.1 The complexity of the algorithm of numeral 2.1 is O((v^2)*e).
2.3 The complexity of the algorithm of numeral 2.3 is O(e)

3.6 2.1 V is the total amount of vertices, and E is the total amount of edges.
2.3 E is the total amount of edges.

4) Practice for midterms

4.1 Table

	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								

4.2 0 -> [3,4]
1 -> [0,2,5]
2 -> [1,4,6]
3 -> [7]
4 -> [2]
5 ->
6 -> [2]
7 ->

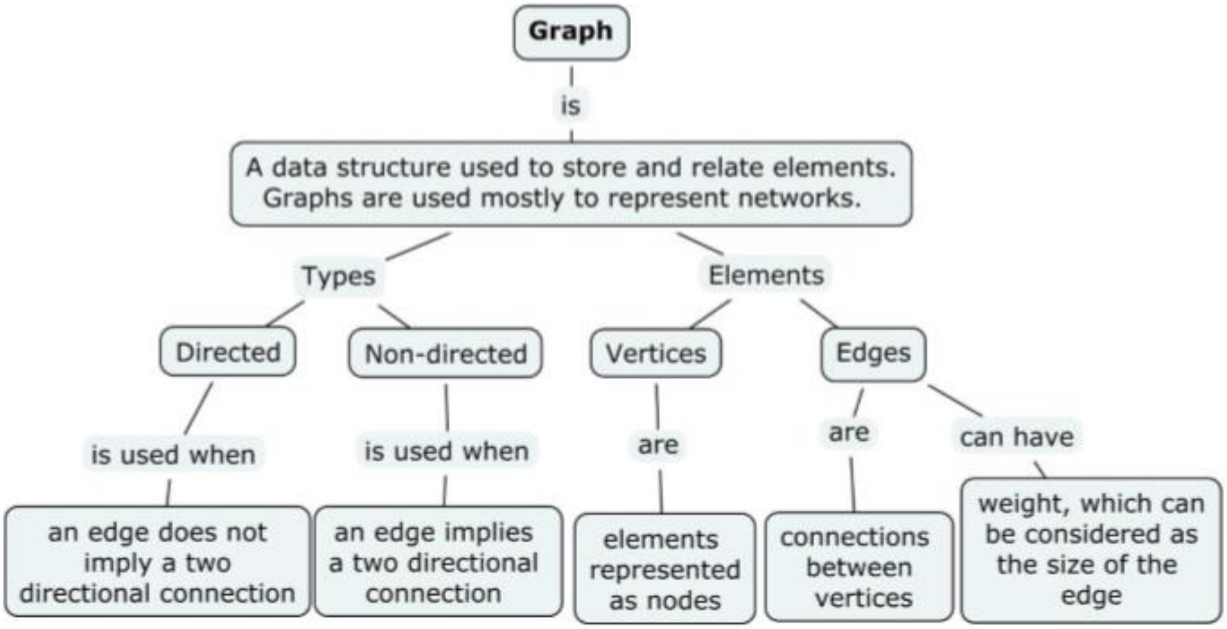
4.3 B) O(n^2)

5) Recommended reading

Title: Chapter 13: Graphs

A graph is a data structure that stores information, but the difference between graphs and other data structures used to store information is that a graph can also relate its elements. A graph has similarities with trees in that a tree can also relate elements, but in a graph hierarchy is not used so it is not important, what is important in a graph is which elements are related and how they are related. A graph has tree important elements, the vertices, the edges, and the weight assign to an edge. The vertices are the elements of the graph, the edges are the connections between vertices and the weight of an edge is a value that tells how “big” the edge is. This weight disappears in problems where it is only needed to know which vertices are related. There also exist two types of graph: the nondirected and the directed. The first indicates that when you can go from A to B, you can also go from B to A, and in the second, this does not happen.

Conceptual map


Graph
is
A data structure used to store and relate elements. Graphs are used mostly to represent networks.
Types
Elements
Directed
Non-directed
Vertices
Edges
is used when
is used when
are
are
can have
an edge does not imply a two directional connection
an edge implies a two directional connection
elements represented as nodes
connections between vertices
weight, which can be considered as the size of the edge

6) Team work and gradual progress (optional)

6.1 Meeting minutes

Member	Date	Done	Doing	To do

Sebastián and Yhoan	01/02/2019	**Think about point 1**		Make a solution to point 1
Sebastián and Yhoan	01/02/2019	**Make a solution to point 1**	Doing analysis of point 2.1	Test for point 1
Sebastián	01/02/2019	**Test for points 1**	Doing analysis of point 2.3	Point 2.1 and 2.3
Yhoan	01/02/2019	**Point 2.1**	Point 2.3	Practice for midterms
Sebastián	02/02/2019	**Point 2.3**	Practice for midterms	Laboratory report
Sebastián	02/02/2019	**practice for mindterms**		Laboratory report
Sebastián	02/02/2019	**Laboratory report**		recommended reading
Yhoan	02/02/2019	**recommended reading**		upload the laboratory

6.2 History of changes of the code

History changes of code		
Version	**Code**	**Status**
1.0	1.0	🟥
2.0	1.0	🟩
1.0	2.1	🟩
1.0	2.3	🟥
2.0	2.3	🟩