Laboratory practice No. 5: Graphs Implementation

Rafael Mateus Carrión
Universidad Eafit
Medellín, Colombia
rmateusc@eafit.edu.co

Daniel Otero Gómez
Universidad Eafit
Medellín, Colombia
doterog@eafit.edu.co

3) Practice for final project defense presentation

3.1 Explanation of code #1

The data structure works with HashMap the class Archivo has a method that reads all the file that wants to be analyzed and then it inserts all the vertex in a HashMap that is being hashed with an id. We decided to use HashMap because after several attempts using TreeSets, LinkedList and ArrayList, HashMap was the only one the ran without exceeding the program memory and it is way more efficient than the others, and it only need to read all the file once, others needed to read them more. Then it identify the arches and creates an object Arco and then with each respective id looks for the vertex that should be added and it instantly adds it to the HashMap.

3.2 Memory consumption with 300,000 vertex

Approximately 670.55 gigabytes.

3.3 Solution

The easiest way to solve this problem was to divide the problem. We decided to use two ArrayList, one for the vertex and another for the archs, as we tried to make a data structure with the class DigraphAl and DigraphAL but this was impossible, because we had a problem in the positions because the ids were not integers, they were long, therefore they could not search a position in the array because the positions on an array are integers.

3.4 Explanation 2.1

The class Graph recieves an integer called vertex that the user enters, for each vertex a LinkedList will be created, each LinkedList will be filled with all the other vertex that each vertex is connected with. Then the method coloring is the one that analyzes if the graph is bicoloreable or not, an array is created and its length equals to the number of vertex and all this array is filled with -1 which means there is not color assigned yet. Then it asings 0 for one color and 1 for another color and proves if they are adyacent or not, if it uses more than two colors then the output will be not bicolorable, otherwise it is coloreable.

3.5 Complexity of 2.1

The complexity is O(nˆ2)

3.6 Variables explanation

n is the number of vertex in the map.

4) Practice for midterms

4.1

	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								

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(nˆ2)