Laboratory practice No. 1: Graphs and search

Juan David Echeverri Villada
Universidad Eafit
Medellín, Colombia
jdecheverv@eafit.edu.co

Juan Sebastián Guerra Hernandez
Universidad Eafit
Medellín, Colombia
Jsguerrah@eafit.edu.co

3) Practice for final project defense presentation

3.1 *First, we read the file and separate each line in three arrays. In the firsts two we save vertex 1 and 2, and in the third one we save the weight of its edge, then, according to its size we create the graph and rename the vertex according to the names given in the text (allowing us to make the conexions only using those names). Finally, we use a for statement to create all the arcs, completing our graphs. We decided to use adjacency list because using adjacency matrix the space complexity would be crazy (more than 83 gb)*

3.2

The implementation of an M_{n*n} array has $O(n^2)$ complexity in memory, with n being the number of rows. If we want to use adjacency matrices for the implementation of a graph with 300 thousand vertexes, defining, for simplicity, that each vertex uses an space of 1 byte, would end in the use of 83.81 GB in memory.

3.3 *We just simply ignore each line that doesn't have the structure of the lines that give us the data needed to create the graph, if we don't find that structure, we go to the next line (solving the identifier´s and title´s problems)*

3.4 *Using an algorithm really similar to BFS, we walk over the dataset, painting the first node with one color, and each of his sons with another one. After that, we paint all of the sons of the sons with the first color, and we repeat that until we detect a collision of colors (a node that was already colored is recolored). If the recursion ends (end condition is same as BFS) without collisions, we can say that the graph is bicolor able, if we found at least one, the graph is not bicolor able*

3.5 *The complexity of the point 2.1 is O(n^2)*

3.6 *n is the number of edges in the graph*

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(n2)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473