Laboratory practice No. 5: Graphs

Alejandro Mc Ewen
Universidad Eafit
Medellín, Colombia
amce@eafit.edu.co

Felipe Henao
Universidad Eafit
Medellín, Colombia
fhenaog3@eafit.edu.co

3.1 The code we made for the first the point was first finding all the possibilities for people picking up other people and getting to the goal in the restrictions. Then we look at all the combinations of this pickups and grab the combination with the minimum number of cars.

3.2 If we represent the map of Medellin of excercise 1 there wouldn't be difference in memory consumption using adyancy lists or adyancy matrices, since it is a complete graph so all nodes have a direct connection with the rest of the nodes.

3.3 Exercise 2.1 we did it preforming a DFS search and coloring the nodes one color and its neighbors the other color. If you ever changed a node of color, then the graph wasn't bicolorable otherwise it was bicolorable.

3.4/3.5

2.1 O(n*e) n = number of nodes and e = number of edges

4.1.1 self.DFS(i + rowNbr[k], j + colNbr[k], visited);

4.2.1

1 -> [0,2,5]

2 -> [1, 4, 6]

3 -> [7]

4 -> [2]

5 -> []

6 -> [2]

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

7 -> []

	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								

4.3 b

4.4.1 ii

4.4.2 i

4.5.1 b

4.5.2 a

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