Laboratory practice No. IV Greedy Algorithms

Daniel Felipe Gómez Martínez
Universidad Eafit
Medellín, Colombia
dfgomezm@eafit.edu.co

Cesar Andrés García Posada
Universidad Eafit
Medellín, Colombia
cagarciap@eafit.edu.co

Daniel García García
Universidad Eafit
Medellín, Colombia
dgarciag@eafit.edu.co

3) Practice for final project defense presentation

3.1 First of all we have a graph and a Boolean array, which will be used to determine if a vertex of the graph has already been visited.
We begin by going through the graph (node to node) and going through the neighboring nodes to the vertex that is being visited at that moment.
Then the following conditionals are analyzed:
1) If the node that is being analyzed is the last node of the graph, this condition is for the stoppage of the algorithm and at the total cost the cost of that node is added to the initial node.
2) Otherwise, if the node next to the node in which it is, is not visited and if the cost between these nodes is less than the cost you have.
If these conditions are met, it is marked as visited the node next and the cost is added.
This process is repeated until the last node of the graph is reached
Finally the cost found is shown, this cost is the minimum.

3.2 For the algorithm to give us a solution, the graph must be a complete graph, because we always want to reach the same starting point and since there is no path between the last nodes of the graph towards the first node, the algorithm would extract an error.

It is not an optimal solution, since all the possible paths are always analyzed, if the nodes of the graph were organized from lowest to highest cost, we would have an optimal solution, since when analyzing a cost, if this does not fulfill the conditions, the others will not they will comply, due to the order.

3.3 A greedy solution to the problem of the traveler agent applied to the delivery of homes in Medellin, would be passing only by the points or nodes where delivery is required and dividing the problem into sub problems, for example, the city is divided into zones.

3.4 The data structure implemented in the solution of the problem is an integer array (int []) where the hours the trucks were delayed were stored. The algorithm only looks for the longest duration of the trajectories both in the morning and in the afternoon, to later find the minimum value of overtime.

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

3.5 Because in order to find the maximum value of an array of n elements and they are two arrays, then the complexity of the algorithm is O (n1 + n2).

3.6
N1 = Size of the arrangement (The number of routes can take a bus in the morning).
N2 = Size of the arrangement (The number of routes can take a bus in the afternoon).

4) Practice for midterms

4.1 i = j

4.2 if (min > adjacencyMatrix[element][i]) {

4.3.1

PASO	a	B	C	D	E	F	G	H
1	**A**	***[20,A]***	∞	[80,A]	∞	∞	[90,A]	∞
2	**B**	[20,A]	∞	[80,A]	∞	**[30,B]**	[90,A]	∞
3	**F**	[20,A]	**[40,F]**	[70,F]	∞	[30,B]	[90,A]	∞
4	**C**	[20,A]	[40,F]	**[50,C]**	∞	[30,B]	[90,A]	[60,C]
5	**D**	[20,A]	[40,F]	[50,C]	∞	[30,B]	[70,D]	**[60,C]**
6	**H**	[20,A]	[40,F]	[50,C]	∞	[30,B]	**[70,D]**	[60,C]
7	**G**	[20,A]	[40,F]	[50,C]	∞	[30,B]	[70,D]	[60,C]

4.3.2 A→B→F→C→D→G (Cost 70)

4.4.1 temp = (temp / 2) + 1
4.4.2 return temp
4.4.3 O(1)
4.5.1 d
4.5.2 I would do a Merge sort to sort the elements from smallest to largest, I would also do a cycle from zero to k-1 and in each iteration I would add the auxiliary array in that position, in the end we would return the auxiliary array that would be the set of k numbers whose sum is minimal. The Merge Sort has complexity (nlogn) and the cycle apart from the method has complexity O (k) that in the worst case would be O (n), we take the largest of these and we have that for the worst case our algorithm has a complexity of O (nlogn).

4.6.1 i + 1

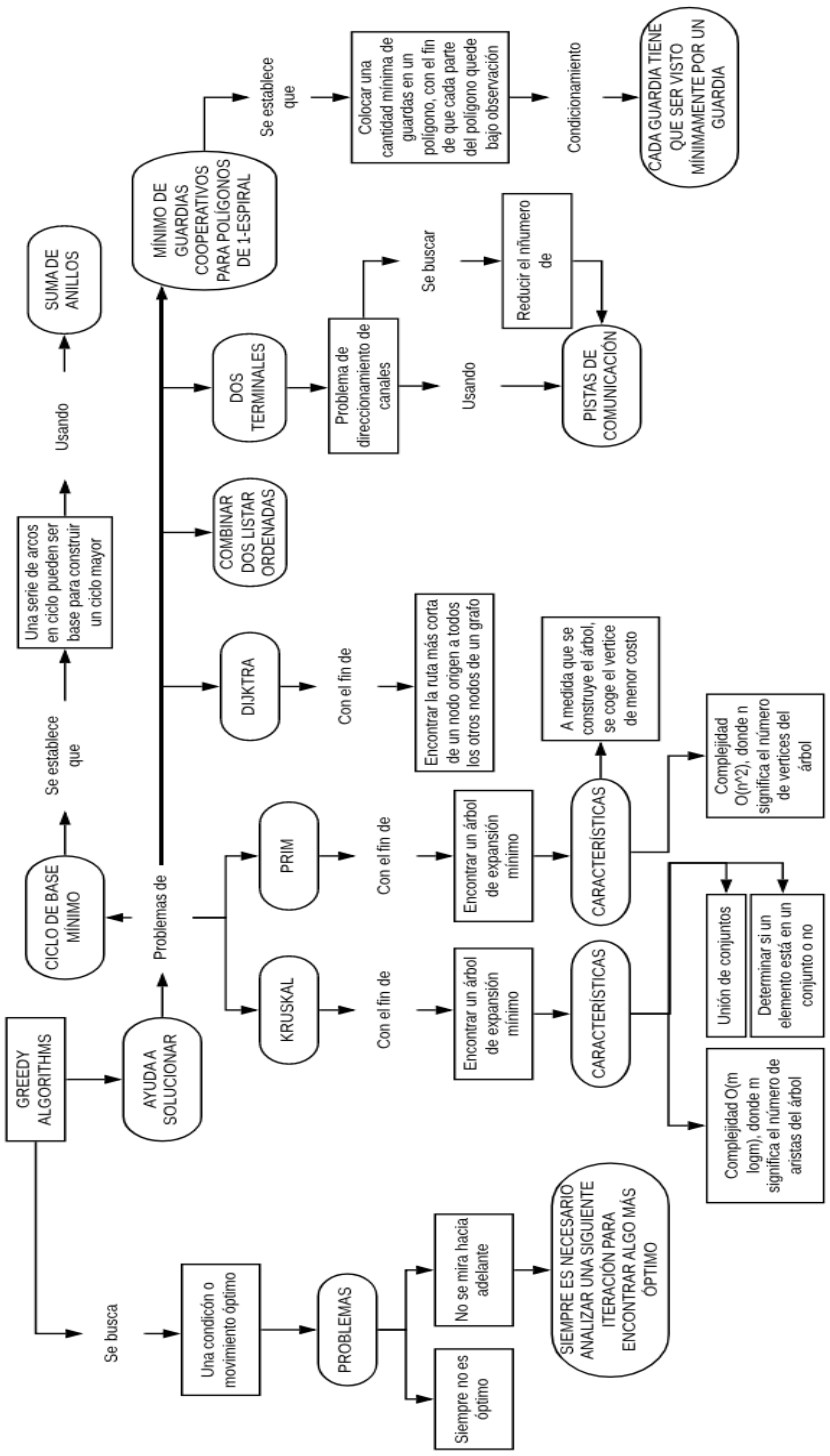
4.6.2 res + 1

4.6.3 i

4.6.4 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

5) Recommended reading (optional)



The conceptual map is attached to github

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

6) Team work and gradual progress (optional)

INTEGRANTE	FECHA	HECHO	HACIENDO	POR HACER
Daniel Felipe Gómez Martinez, Cesar Andres Garcia Posada Y Daniel Garcia Garcia	11/04/2019		Reading and understanding the laboratory.	Methods to work with graphs
Daniel Felipe Gómez Martinez	12/04/2019		Reading the file ".txt" and the Backtracking_Digraph class. (For point 2).	Laboratory Report
Daniel Garcia Garcia	12/04/2019		Practice for midterm (Point 4)	Practice for midterm (Point 4)
Daniel Felipe Gómez Martinez	13/04/2019	Reading the file ".txt" algorithm	Elaborating minimum cost algorithm (For point 2).	Laboratory Report
Cesar Andres Garcia Posada	14/04/2019		Algorithm to find the shortest path between two points using greedy algorithms (Point 1)	Algorithm to find the shortest path between two points using greedy algorithms (Point 1) Laboratory Report
Cesar Andres Garcia Posada	15/04/2019	Algorithm to find the shortest path between two points using greedy algorithms.	Laboratory Report	
Cesar Andres Garcia Posada	15/04/2019	Laboratory Report		

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

Daniel Garcia Garcia	14/04/2019	Practice for midterm (Point 4)	Laboratory report	Laboratory report
Daniel Felipe Gómez Martinez	13/04/2019	Elaborating minimum cost algorithm (For point 2).	Laboratory report	
Daniel Felipe Gómez Martinez, Cesar Andres Garcia Posada Y Daniel Garcia Garcia	15/04/2019	Laboratory report	Optional points for more learning.	

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