ESTRUCTURA DE DATOS 2
Código ST0247

Laboratory practice No. 4: Greedy Algorithms

Alfredo José Ospino Ariza
Universidad Eafit
Medellín, Colombia
ajospinoa@eafit.edu.co

Jonatan David Acevedo López
Universidad Eafit
Medellín, Colombia
jdacevedol@eafit.edu.co

3) Practice for final project defense presentation

3.1 The data structure used was a List (an "*Array List"* in this case), and the focus of the algorithm is just finding the lowest cost while traversing the map (in this case a Graph) without going back into any of the cities, by doing a nested loop that makes sure the algorithm doesn't traverse into cities it's already been to. At the end it just adds the last travel which is going back to the deposit.
3.2 No, it doesn't always give the least costing solution, the algorithm needs to have a map, the weights of the paths (meaning how much it costs for the truck to go through that path it being gas, time, etc.) it is coursing through, and the amount of packages it's going to be delivering that day, and the packages it has delivered that day already. Because without them the information to continue would be incomplete, as an example; how would the messengers or messenger, know how many packages they had left to deliver? Or maybe how would they know where would they need to go?
3.3 Using our solution to the problem it would only really need to use the map of Medellín, implement it as graph, and know the lengths of the travels the truck it´s going to make, for example, from Bello to Envigado, Envigado to La Estrella, La Estrella to Sabaneta, and things like that. You would only need to traverse the spots you have to deliver something in, not the complete graph. In order to know the distances to traverse the city, I would just say use something like Google maps or Waze, so the truck driver can have more or less an idea according to the kilometers between spot A and spot B, but we can't really think of any other way this could be achieved, unless the graph of Medellín came with the distances already, but that seems improbable.
3.4 The data structure would be basically and array to store the variables, n, d, r,
the algorithm receives a string with spaces, that string separates it and converts them to integers (n, d, r) respectively. Later in the process of the morning and the afternoon (tours) the same thing is done to see if the values that are given in those linear is greater ad, if this is fulfilled then the difference between d and the line multiplies it by r (and so the extra value is extracted for each hour) at the end shows them. If you get to enter a line "0 0 0" you will leave the cycle and finish the process.
3.5 O(n)
3.6 n is the number of routes that are made, since the cycle that was done that adds the morning and afternoon routes, depends on this.

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

4) Practice for midterms

4.1 *i++*
4.2 *min>adjacencyMatrix[element][i]*

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