ESTRUCTURA DE DATOS 2
Código ST0247

Laboratory Nro. 4
Greedy Algorithhms

David Alzate Cardona
Universidad Eafit
Medellín, Colombia
Dalzatec1@eafit.edu.co

Laura Alzate Madrid
Universidad Eafit
Medellín, Colombia
lalzatem@eafit.edu.co

3) Questions of sustainability of projects simulated

3.1 The data structure used in this algorithm is an arraylist. This algorithm works with complete graphs, which starts from a node that we call the source node (in our test case is the node 0), and then moves to the node that has the lowest cost, always asking if it has already been visited, if not She goes to visit him and with an array of Booleans sends you "true". The algorithm does this with all the nodes until you visit them all, without repeating any, and returning to the origin node.
3.2 Using a greedy algorithm, in this algorithm it is not possible to always deliver the most optimal solution. The graph must be a complete graph, otherwise it is impossible to give a solution since it couldn't visit all nodes and the node of origin, and this is one of the conditions that traveler must obey in the algorithm of the agent.
3.3 This algorithm can be modified to the problem of delivery services; the path must necessarily pass through each node where there is a delivery to the product.

3.4 We use Arraylist to save the extra money from each day of work, the algorithm analyzes the hours that makes each bus driver in the morning and in the evening, then it makes an addition to those hours and that result are the amount of hours he works in a certain day, then it compares the past result with the hours each bus driver should work in a day and what is left extra from that comparison are the hours that the company must pay him, and then the algorithm makes an addition from the total from a certain day and throws as a result what the bus driver must be paid for
3.5 O(n).
3.6 N is the number of bus routes

4) Practice for midterms

4.1 *i=j*
4.2 *min > adjacencyMatrix[element][i]*
4.3
4.3.1

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

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	70,D	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
8	E	20,A	40,F	50,C	∞	30,B	70,D	60,C

4.3.2 A→B→F→C→D→G

4.4

4.4.1 *Temp/2*
4.4.2 *Temp + minimo*
4.4.3 *b) O(1)*

4.5

4.5.1 *d) The couple 2 is right and its justification is true*

4.5.2 We organized the arrangement with the method array.sort() from the fewer value to the biggest value and then we would make an addition between the first k numbers to be able to get the lowest aggregate, all this would be done in O(nlogn)

4.6

4.6.1 *i+1*
4.6.2 *res + 1*
4.6.3 *i*
4.6.4 *x_t = {1,0,0,4,0,0,0,0,0,0,11}, k_t = 3. Your output is 2.*

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473