Laboratory practice No. 4: Greedy Algorithms

Camila Barona Cabrera
Universidad Eafit
Medellín, Colombia
cbaronac@eafit.edu.co

Felipe Sosa Patiño
Universidad Eafit
Medellín, Colombia
fsosap@eafit.edu.co

3) Practice for final project defense presentation

3.1 The data structured used at this problem is an arraylist, which stores the successors of a graph and an array that Will allow us to verify if we have already visited a node or not, so that when we visit the adjacent ones, we proceed to verify if it has not been visited and if the cost between the inicial node and the successor is les tan the mínimum, if these conditions are right, we obtain a new mínimum cost and we continue with the procedure with the next node.

3.2 No, with this data structured won't always give us the optimal solution. The graph must be complete to show a posible solution because we can found unreachable nodes and could be unvisited.

3.3 The greedy algorithm can be used in the problem of delivery services, only visiting the nodes where we must deliver a product.

3.4 The data structure that we are using is a simple array for saving the information of the durations of the morning routes and other array for saving the durations of the afternoon routes. We don't use lists because we already know how many routes we are working with since we know the number of drivers. The Algorithm takes those arrays, the maximum duration for the drivers, and the rate for paying per extra hour, after that, with help of a function of the library "Arrays" named sort, we sort the arrays and with just one loop, a for loop that roams through both arrays at the same time but the afternoon array is traveled upside down for do equilibrate couples of routes of the morning and of the afternoon.

3.5 The complexity of the algorithm, programed at the method "minimumHoursValue", has to consider, the complexity of sorting both arrays, at the api of java, we find the complexity of that method is O(nLog(n)) as you can see in the next Image from de api:

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

sort

```
public static void sort(int[] a)
```

Sorts the specified array into ascending numerical order.

Implementation note: The sorting algorithm is a Dual-Pivot Quicksort by Vladimir Yaroslavskiy, Jon Bentley, and Joshua Bloch. This algorithm offers O(n log(n)) performance on many data sets that cause other quicksorts to degrade to quadratic performance, and is typically faster than traditional (one-pivot) Quicksort implementations.

Parameters:

```
a - the array to be sorted
```

Next, we have a for loop which complexity is O(n), so, the entire complexity is the biggest, which is O(nLog(n)).

3.6 n is the number of routes and at the same time, is the number of drivers. The other letters that join in the algorithm are d and r, which are, the maximum duration per day for each driver, and r is the rate for paying per extra hour.

4) Practice for midterms

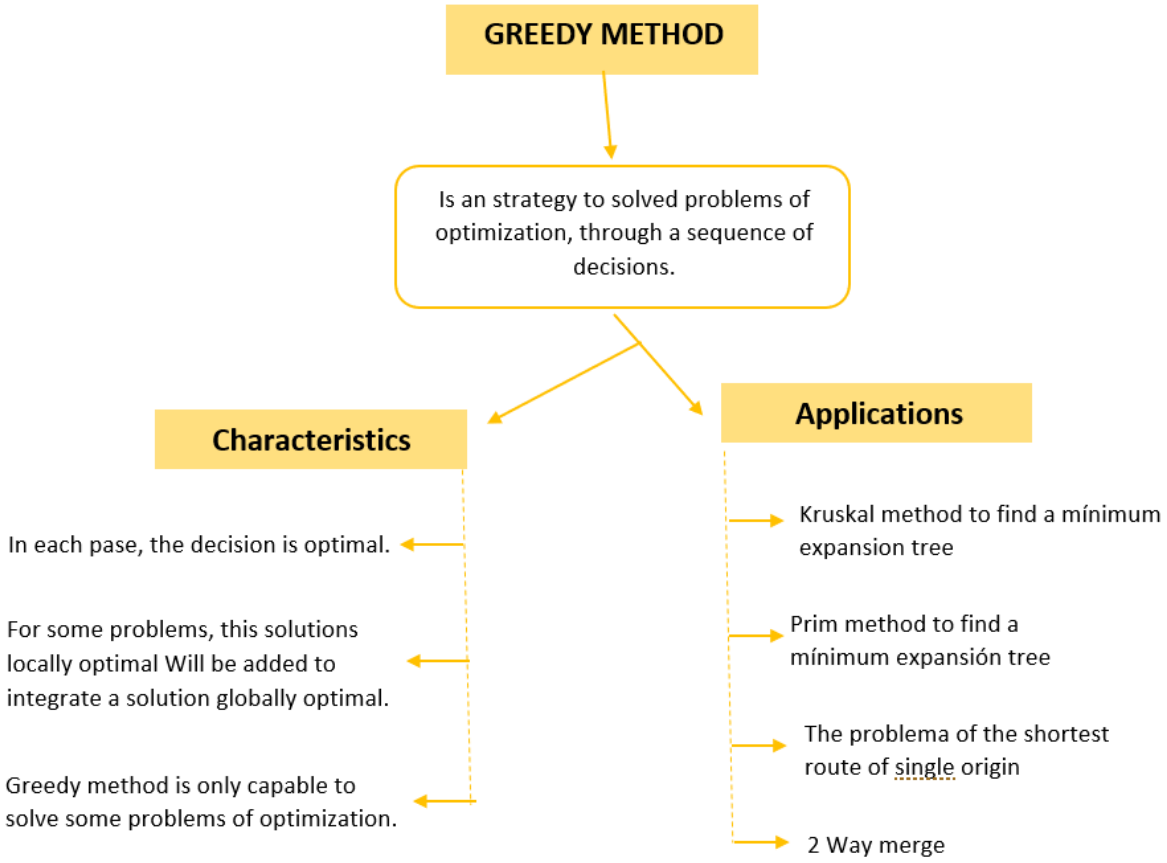
4.1 i=j;
4.2 Line 18: min>adjacencyMatrix[element][i]
4.3.1

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	C	20,A	∞	70,F	∞	30,B	90,A	∞
4	D	20,A	40,F	50,C	∞	30,B	90,A	60,C
5	E	20,A	40,F	50,C	∞	30,B	70,D	60,C
6	F	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	H	20,A	40,F	50,C	∞	30,B	70,D	60,C

4.3.2 A -> B -> F -> C -> D -> G, this way has a cost of 70.
4.4.1 Line 10: temp/2
4.4.2 Line 11: temp + mínimo
4.4.3 Complexity: O(1)
4.5.1 d.
4.5.2 The algorithm for this problem will consist in order the elements in order from lowest to highest and sum the k numbers stored in an array. The complexity is: O(n) for the cycle to order the numers of the array.
4.6.1 i+1
4.6.2 res+1
4.6.3 i
4.6.4 2

5) Recommended reading (optional)

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)

Member	Date	Done	To do
Camila Barona	12/04/2019	Search a solution to point 1	Implement a solution to point 1
Camila Barona	13/04/2019	Implement a solution to point 1	Search a solution to point 2
Felipe Sosa	12/04/2019	Search a solution to point 2	Implement a solution to point 2
Felipe Sosa	14/04/2019	Implement a solution to point 1	Make the laboratory report
Camila Barona and Felipe Sosa	14/04/2019	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

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