

Laboratory practice No. 4: Greedy Algorithms

Miguel Sosa Villegas
Universidad Eafit
Medellín, Colombia
msosav@eafit.edu.co

Sergio A. Córdoba Muriel
Universidad Eafit
Medellín, Colombia
sacordobam@eafit.edu.co

Miguel Jaramillo Arenas
Universidad Eafit
Medellín, Colombia
mjaramil20@eafit.edu.co

3) Practice for final project defense presentation

3.1 We used a linked list, because it's easier to handle than the adjacency matrix, and we used the dijkstra algorithm, which consists of going to the shortest path between the successors of the nodes and so on until it reaches the destination.

3.2 It doesn't always get the optimal solution, because it chooses the shortest path among all the successors without considering the successors of the successors, so it's not the optimal path always, and the graph needs to have at least one possible solution for it to work.

3.3 It could be possible, but you need to go between most of the nodes of the graph because the only way of going to one location to another is a street, so you need to go between streets to arrive at the destination, so the distance between two nodes is measured by the length of the street.

3.4 The data structure we used was the array, to represent the duration of the routes in the morning and in the afternoon. The algorithm creates the arrays and then it assigns a pair of routes (one of the morning and one of the afternoon) to each driver, following the rule of picking the highest duration of the morning with the lowest duration of the afternoon. Then in the end it verifies if there are any extra hours and then it finally multiplies it by the rate, and returns the number given.

3.5 The complexity is $O(n)$, that is given by the number of drivers or routes.

3.6 N is the number of drivers

4) Practice for midterms

4.1 Activities: $i = j$

4.2 Travel Agent: $\min > \text{adjacencyMatrix}[\text{element}][i]$

4.3 Optional

4.4 Vitamins:

4.4.1 $\text{temp}/2$

ESTRUCTURA DE DATOS 2

Código ST0247

4.4.2 **temp + mínimo**

4.4.3 $O(1)$

4.5 Optional

4.6 The King Arrival:

4.6.1 **i + 1**

4.6.2 **res + 1**

4.6.3 **i**

4.6.4 Output: 2

5) Recommended reading (optional)

Summary: Greedy Method

It's a strategy to solve optimization problems. It will be assumed that it is possible to solve a problem through a sequence of decisions. In every step, the decision is the most optimal, it is important to specify that this type of method is capable of solving some optimization problems.

In case the locally optimal decisions do not result in a globally optimal decision, the greedy method is still a good option to implement because at list it produces an acceptable solution.

In other words, the greedy method consists in trying all the possible solutions in order to select the one(s) that is most optimal as a global solution to each problem.

Applications:

- **Kruskal's method for finding a minimum spanning tree:** for Kruskal the minimum expansion trees are defined on graphs.
- **Prim's method for finding a minimum spanning tree:** something important about this method is that it can start in any vertex.
- **The problem of the shortest route of single origin (Dijkstra):** it consists in finding the shortest way from a determined vertex to all the others.
- **2-way merge:** merge two sorted lists.
- **Minimum cycle basis:** 2 cycles could merge with each other.
- **Minimum number of guards for a polygon:** the problem is to find a way in which each guard could see every part of the polygon.

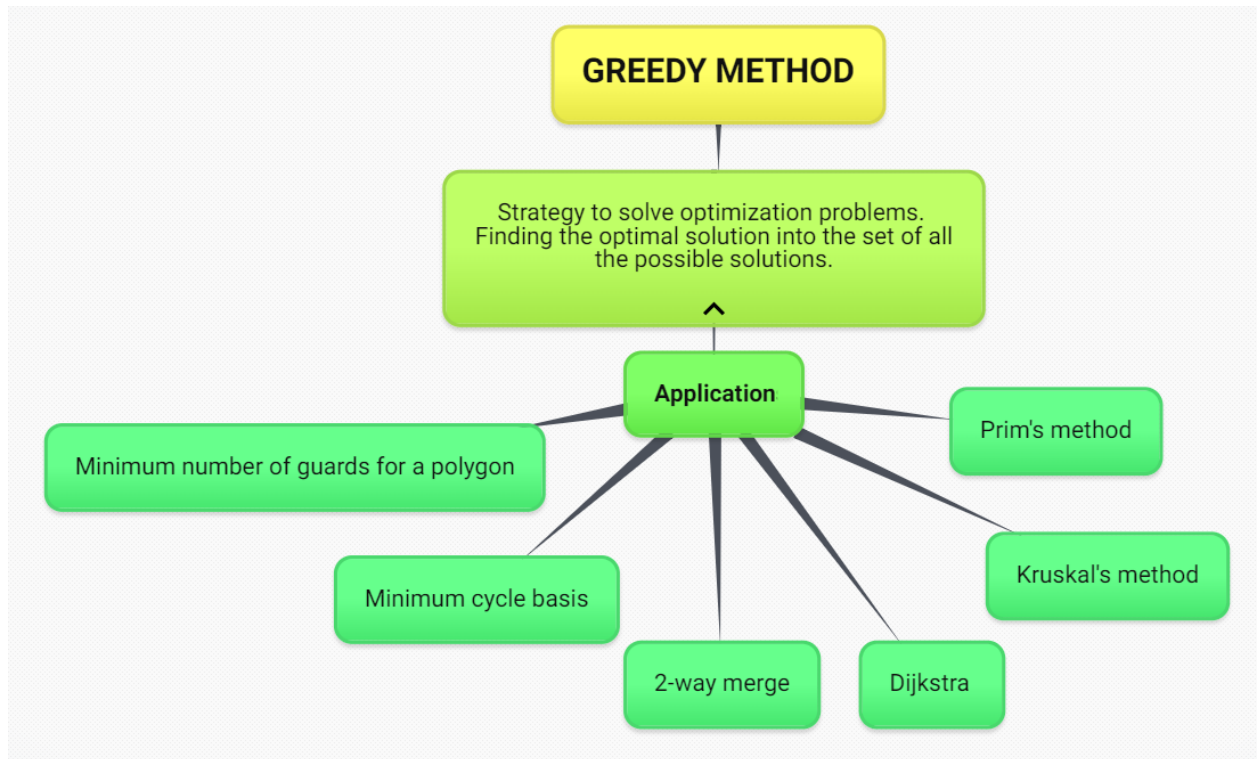
PhD. Mauricio Toro Bermúdez

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

Conceptual Map

















6) Teamwork and gradual progress (optional)

6.1 Kanban Board

ESTRUCTURA DE DATOS 2

Código ST0247

Punto 2.1  miquel jaramillo March 27, 2022 8:07 PM	
3.6  miquel jaramillo March 27, 2022 9:45 PM	
3.5  miquel jaramillo March 27, 2022 9:45 PM	
3.4  miquel jaramillo March 27, 2022 9:45 PM	
Punto 4.1  sergio March 27, 2022 3:45 PM	Punto 3.1  Miquel Sosa March 27, 2022 5:08 PM
Punto 4.2  sergio March 27, 2022 3:45 PM	Punto 3.2  Miquel Sosa March 27, 2022 5:28 PM
Punto 1.1  Miquel Sosa March 27, 2022 4:05 PM	Punto 3.3  Miquel Sosa March 27, 2022 5:30 PM
Punto 4.4 O(1)  sergio March 27, 2022 4:40 PM	Mapa Conceptual  sergio March 27, 2022 7:31 PM
Punto 4.6  sergio March 27, 2022 4:56 PM	Resumen Lectura  sergio March 27, 2022 7:31 PM

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

6.2 History of changes of the code

commit 6693dd00edc20744456dcd4099de0434db3dfa6a (HEAD -> master, origin/master, origin/HEAD)

Author: mstermigol <85334763+mstermigol@users.noreply.github.com>

Date: Sun Mar 27 21:30:30 2022 -0500

Add files via upload

commit bea02cf16d307edbd7c6e8678c8a8eb2e857b803

Author: Miguel S <85181687+msosav@users.noreply.github.com>

Date: Sun Mar 27 17:04:04 2022 -0500

Add files via upload

6.3 History of changes of the report

