CARPOOLING

(Reduce congestion and pollution in the city by transporting several people in a car that travels the shortest distance from the starting point to the university)

Santiago Moreno Rave	Valeria Suarez Mejia	Mauricio Toro
Universidad Eafit	Universidad Eafit	Universidad Eafit
Colombia	Colombia	Colombia
smorenor@eafit.edu.co	vsuarezm@eafit.edu.co	mtorobe@eafit.edu.co

RESUME

As much as in the Eafit university as in the city of Medellín, there is a problem of air pollution and people are expected to share their private vehicle by making a route in which they can fill the quota of their vehicle with other people, now in the part of algorithms the problem is that you can travel a graph (the city of Medellin) in the minimum time from any node of the city to the university Eafit, some related problems are those like the shape of the city, its roads and traffic.

Keywords

Carpooling, vehicle with maximum capacity, route analysis, shorter routes, minimum route through the city.

Keywords of the ACM classification

Data Structures Graph and tree search strategies Graph algorithms análisis Hash-table representations [Data Storage Representations] Linked representations [Data Storage Representations] Shortest paths

1. INTRODUCTION

Transportation is something that all citizens (in this case students) use to reach a destination. For people who need to get to the university campus, carpooling can be a good alternative, since the idea of this project is to find the shortest routes where people are picked up to take each car to its maximum capacity.

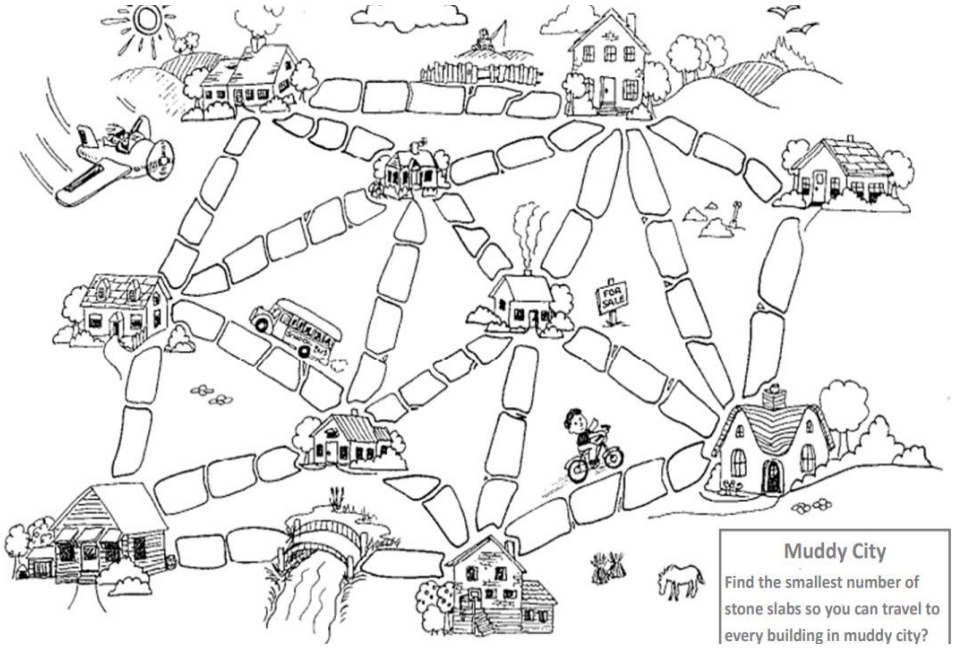
2. PROBLEM

It is the way in which we must travel the graph (map of Medellín) with the minimum cost, it would be solved for a more effective functionality of the system that we want to build for the solution of the problem. By solving this problem we would be giving the city a new way of transporting itself, becoming a sustainable medium and increasing the ease of mobilization in the city.

3. RELATED JOBS

3.1 A Muddy City

There is a graph in which there are no streets and enough streets must be paved so that it is possible to travel from any house to another house, it is possible to travel through houses. Paving must be done at a minimum cost. This is going to be done with paving slabs, so the question is, what is the smallest number of slabs needed to create the streets at the minimum cost?

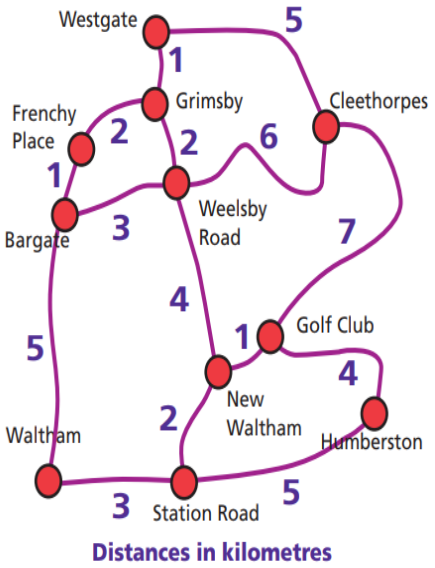


visualización del problema

3.2 Cable Connections

Can you link all the nodes using the shortest cable?

The challenge is to know how to connect the cables in all the places of the map, for a cable television service. Using more cable than necessary can generate high costs, so it is necessary to keep the total cable length as short as possible.



visualización del problema

3.3 Connected

Blizzard has toured the entire state, covering all roads with 40 inches of snow. To maintain food distribution, the roads between the main cities must be cleared, so that each of the main cities can be reached. But getting rid of snow on a highway is expensive: $ 1000 per mile. You must design a plan that keeps all major cities connected through clear roads, with the cost of cleaning as small as possible. You are given the list of distances (in miles) of direct connections (ie, no other important route is passed) between the main cities. You must perform the mathematical calculations and choose which of those connections should be eliminated.

3.4 The Santi Road

For more than a thousand years the road to Santiago has been traveled as a pilgrimage to the sanctuary of the apostle Santiago el Mayor, located in the city of Santiago de Compostela in Galicia (Spain). His tomb was discovered in the year 812 on the sacred mountain of Libredón, and since then thousands of pilgrims come to visit it every year. The pilgrimage to Santiago is considered the cultural act and Jugueteando con grafos R. Núñez, J. Núñez, E. Paluzo, E. Salguero Number 46 - June 2016 - Page 192 most outstanding religious of the Middle Ages, recently recognized by the European Parliament as the First European Cultural Itinerary and by UNESCO as a World Heritage Site (Mielnikov, 2011). Regarding this road to Santiago, two well-known scientists came up with the following problem: Descartes (1596-1650) and Newton (1643-1727), mathematician and physicist, respectively, and both very religious, decide to walk the Camino de Santiago starting from Seville. Descartes proposes to make an alternative route to the conventional and Newton, which is somewhat manic, imposes that it would only pass through certain cities (Córdoba (2), Ciudad Real (5) Madrid (12) and Valladolid (19)) and also wants the route is as short as possible. Given this, Descartes intends to see all the existing roads between the cities chosen by his fellow maniac.

Figura 4

We see that a multigraph has been constructed from the different cities (Fig. 4). If we go from Sevilla to Santiago passing only once for each vertex, we build a tree graph with all the possible routes (Fig. 5)

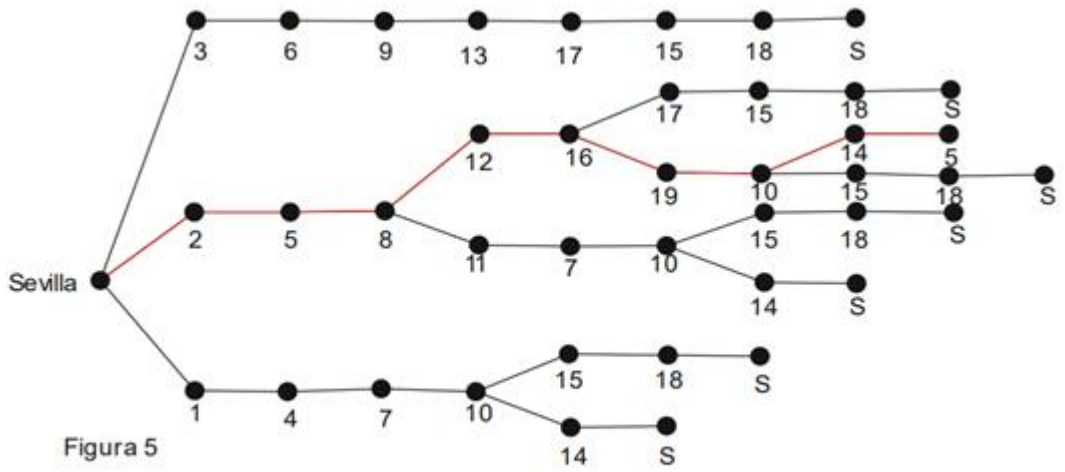


Figura 5

Each of the branches of the tree graph is a route and the number of vertices of that branch tells us how many cities make up the route. To solve this problem we must find a path in this graph that contains the vertices 2, 5,12 and 19. Also, since it must be as short as possible, we must choose the branch of the tree graph that contains the least vertices. We conclude that the road that pilgrims should choose is the third road (red road Fig. 5) that arrives in Santiago.

4. TITLE OF THE FIRST DESIGNED SOLUTION

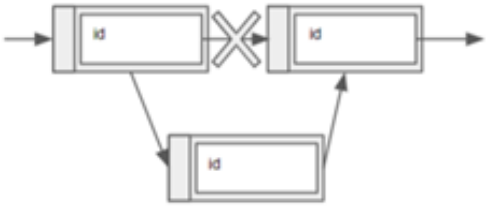
"Carpooling picks up and carries"

4.1 Data structure

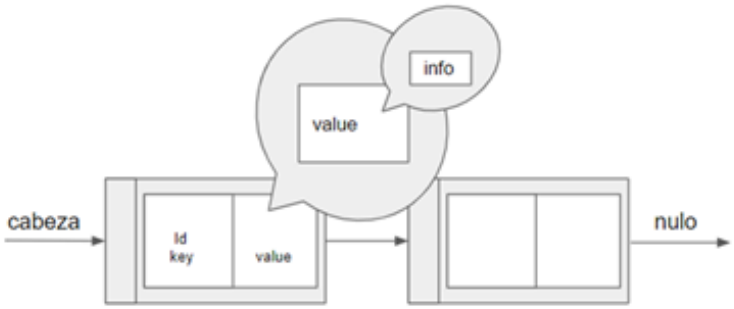
The first version of data structures will be divided first into storage of information, for this a Hashmap will be used, the key of this will be the node, in the value of each position of the HashMap, there will be a LinkedList of pairs, which will contain in the first position of the "pair" the personal information of the user in the form of object "person", and in the other position of the "pair" there will be a list of the successors (the successor has its own method to obtain the weight of the arc). For the second part we will use the algorithm or data structure of Dijkstra for the complete route of the graph and find the most optimal route for a person (the driver) can reach x number of people, taking into account the maximum quota of the vehicle.

4.2 Data structure operations

Add in the LinkedList:



getValue of HashMap:



4.3 Design criteria of the data structure

We designed this way because it is a very graphic way to represent the solution, since it can be interpreted as a type of hierarchy when linking all these structures (refers to the LinkedList, HashMap, etc.). We also designed it that way, because it allows us to place or insert elements without having to have any specific order, the inserted nodes since the HashMap uses "keys" that allow me to access them without any type of order.
The efficiency of the path of the nodes will be optimal, because we are ensuring that no node is left without visiting, taking into account the minimum cost.

4.4 Complexity Analysis

Method	**Complexity in the worst case**	**Complexity**
Search in HashMap	O(n)	O(1)
Search in LinkedList<Pair>	O(n)	O(n)
Remove items from the HashMap	O(n)	O(1)
Verify key of a HashMap	O(n)	O(1)

4.5 Algorithm

The algorithm will be based on doing a Quicksort to the arrangement of the weights of the successors of each node, then what is done is to choose the first position (that with the Quicksort we ensure that it is the minimum cost), then we will go to the node chosen successor and the same procedure is done until reaching the destination. Each node that is going to be changed will be marked as visited, in order to have a check that the successors of the same node are not reviewed twice.

4.6 Calculation of algorithm complexity

Sub problem	**Complexity**
Create graph	O(n)
Go through the graph	O(n^2)
Complexity total	O(n^2)

4.7 Algorithm design criteria

We decided to do it this way because of the effectiveness of the graph and the way in which the data are handled (of the people involved in carpooling).
The HashMap is used because the complexity of the algorithm decreases when going from one node to another due to the fact that with the Key of each position, it goes directly to the node.
The LinkedList Pair, we use it because it allows us to store two values, in which other types of data structures can also be stored, which helps us to be more versatile when it comes to storing and linking the data.
The LinkedList is used because the complexity of this structure is lower than it is a normal arrangement.

4.8 Execution Times

	Data set 1	***Data set 2***	***Data set 3***
Best case	**2 ms**	**30 ms**	**62 ms**
Average case	**1 ms**	**100 ms**	**10 ms**
Worst case	**1 sg**	**2 sg**	**4 sg**

4.9 Memoria

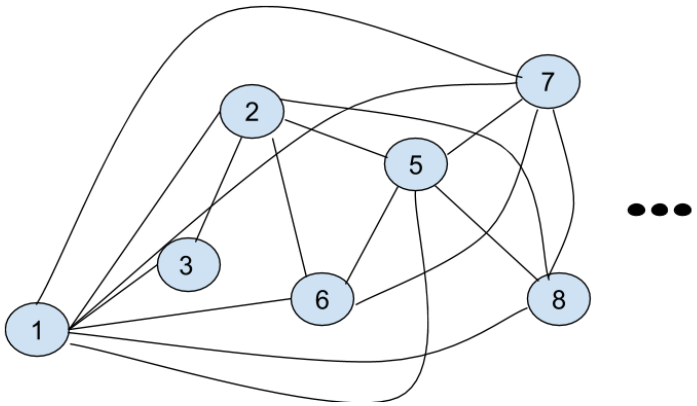
	Data set 1	*Data set s 2*	*Data set 3*
Memory consumption	123 MB	284 MB	365 MB

5. TITLE OF THE DESIGNED SOLUTION

"Carpooling picks up and carries"

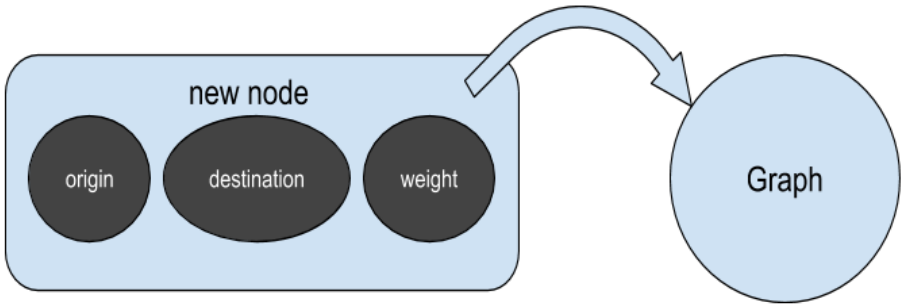
5.1 Data structure

The first phase for the realization of the project is the storage of the data, this will be done with the help of the implementation of a complete graph (a complete graph is used because it represents the map more easily and due to the fact that they are all connected with all). Thanks to this graph we will have access to all the nodes and the weights of all the nodes. We use ArrayList to store the passengers of each car (which must not exceed 5 passengers), and we use another ArrayList ArrayList to store all the cars that are validated.

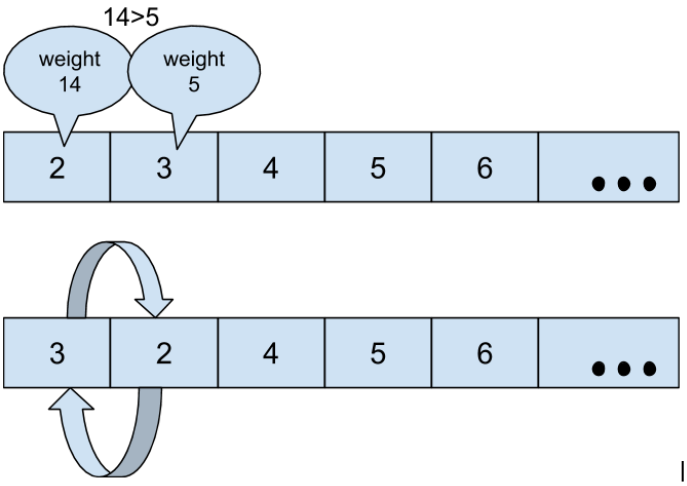


5.2 Data structure operations

addArc (add a node with its respective weight to the graph)



ArrayList ordering (from least to greatest with respect to weight)



5.3 Design criteria of the data structure

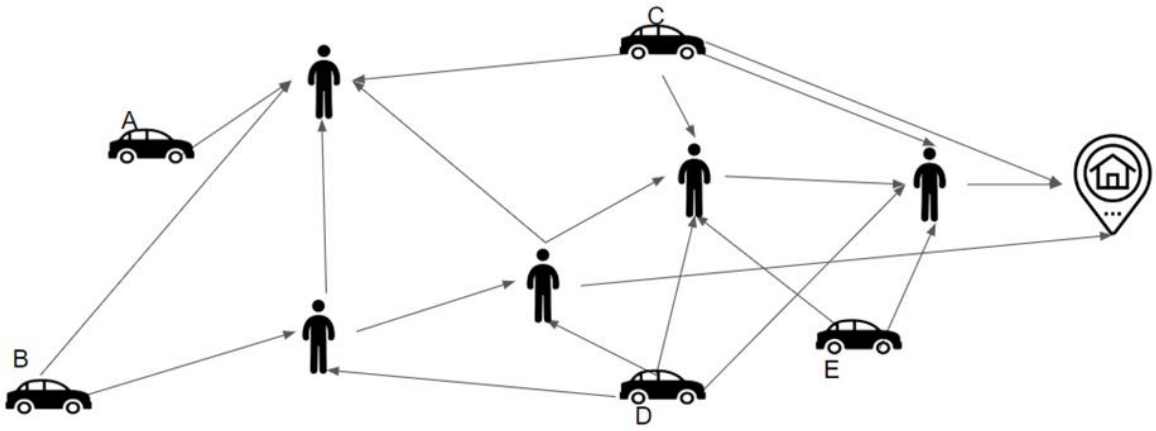
We designed this way because it is a very graphic way of representing the solution, since it can be interpreted as a type of hierarchy when linking all these structures (it refers to the implementation of the complete graph and to the use of ArrayList, etc.). We also design it that way, because it allows us to place or insert nodes in an organized way depending on the weight that each node has with respect to the other node. In addition, the algorithm starts from the farthest node to the destination node.
The efficiency of the path of the nodes will be optimal, because we are making sure that no node is left without visiting, taking into account the restriction of the p.

5.4 Complexity Analysis

Method	**Complexity in the worst case**	**Complexity**
Add node	O(n)	O(n)
Linear ordering	O(n^2)	O(n^2)
Verify driver	O(n)	O(n)
Assign to the car	O(n)	O(n)

5.5 Algorithm

The algorithm will be based on ordering an ArrayList (which will contain the direct nodes to the university) from least to greatest (with the algorithm of Linear ordering that is of complexity n ^ 2) from the weights of each node, it goes to take the one that has more weight as the driver, since it means that it is the furthest from the university in relation to other nodes. Therefore it will be traveling from the farthest place, to the university, filling ArrayList as possible with 5 passengers, that ArrayList will be taken as a car, and the same procedure would be done for the next nodes of the last chosen . The duration of the route, which is always less than or equal to p.



5.6 Calculation of algorithm complexity

Sub problem	Complexity
Create graph	O(n)
Order	O(n^2)
Get weight	O(1)
Complexity Total	O(n^2)

5.7 Algorithm design criteria

We decided to do it this way because of the effectiveness of the graph and the way in which the data are handled (of the people involved in carpooling).
The ordering algorithm is used for the optimization of the way of distributing the cars, since knowing which are the closest to some nodes and these do not add time to their journey to the university (refers to the p) can be carried by them, apart they will be removed from the main list of direct nodes to indicate that they already have auto assigned or are drivers.

5.8 Execution Times

	Data set 1	***Data set 2***	***. Data set 3***
Best case	**2 ms**	**10 ms**	**15 ms**
Average case	**2 ms**	**14 ms**	**40 ms**

Worst case	**28 ms**	**64ms**	**154 ms**

5.9 Memory

	Data set 1	***Data set 2***	***Data set 3***
Memory consumption	12 MB	24 MB	36 MB

6. Conclusions

- Having everything ordered in a specific way it is easier to have a clear answer, for them the ordering algorithms are so useful when handling problems of this type.

- Subdividing the problems that arise when solving the larger objective can obtain a quite optimal result, that is why it is better to divide and conquer!

-Working with a complete graph is much easier to use data structures as matrices in this case a linked list of linked lists.

6.1 Future works

We would like to improve the complexity for a faster handling of the data, since in a much larger cities a greater speed would be needed.
Also for other problems such as the delivery of pizzas, or some home app, or some service that will need more speed.

REFERENCES

1. Computing At The School, Toy Problems For The Real World, https://www.computingatschool.org.uk/data/tft/02/04Activity_Toy_Problems_Real_World.pdf

2. Chegg Study, For a given real-world problem like the one below, which, https://www.chegg.com/homework-help/questions-and-answers/problem-9-given-real-world-problem-like-one-algorithm-use-solve-would-map-input-informatio-q17272505

3. Rocío Núñez Santiago, Juan Núñez Valdés, Eduardo Paluzo Hidalgo, Elena Salguero Quirós, Jugueteando con Grafos, http://www.fisem.org/www/union/revistas/2016/46/10_21-401-1-ED.pdf