Laboratory practice No. 5: Divide to Conquer and Dynamic Programming

DANIEL GARCÍA GARCÍA	DANIEL FELIPE GÓMEZ MARÍNEZ	CESAR ANDRES GARCÍA POSADA
Universidad Eafit Medellín, Colombia dgarciag@eafit.edu.co	Universidad Eafit Medellín, Colombia dfgomezm@eafit.edu.co	Universidad Eafit Medellín, Colombia cagarciap@eafit.edu.co

3) Practice for final project defense presentation

3.1

To solve the problem of the traveling agent through Held-Karp we have a cost matrix, which is composed of the whole power of the vertices, divided two because as we have a start node and one end, the power set containing that node. We also have a binary mask that corresponds to each set power to calculate the cost, this to facilitate the work with power sets. This cost is calculated as follows:

- If the power set has only one node, the cost will be the cost between the initial node a is node of the power set and this to the arrival node.
- If the power set has two nodes, the cost will be the minimum between the cost of the initial node to the first node of the power set and this to the final node and of the initial node to the second node of the power set and to the final node.
- If the power set has three nodes, the costs corresponding to each one are searched in the matrix, this is making combinations to find the cost of the way in which the power set has only two nodes.

In the end, the minimum cost will be the cost that is in the last position of the matrix.

3.2

Taking into account that the complexity of the algorithm is n^3 where n means the number of nodes, having a graph with 50 nodes, we have 125,000 operations

3.3

The data structure that we implemented for point 2.1 was a HashMap which has as key an entro and as content an array of integers. Each key contains an arrangement of length two which has in the first position the row which is the toxic and the second position the column in which the toxic is located.

Through the HashMap we can find the shortest route from one point to another so that Karolina (which is the starting point and end of the route) can go through the toxics and return to its initial position.

To solve a graph of 50 it would be necessary to solve 2500 operations.

3.4

The data structure that we implemented for point 2.1 was a HashMap which has as key an entro and as content an array of integers. Each key contains an arrangement of length two which has in the first position the row which is the toxic and the second position the column in which the toxic is located. Through the HashMap we can find the shortest route from one point to another so that Karolina (which is the starting point and end of the route) can go through the toxics and return to its initial position.

3.5

Because we traverse the graph n times to find the shortest path from a fixed point to a toxic (another vertex) it is traversed n * (n-1) so for the but of the cases it is n^2

3.6

K = Integer Arrangement (initial and final position of the route).
n = It is the amount of toxics that you have to collect (vertices).
n-1 = The number of vertices remaining when a shorter path from K is chosen to one of the vertices.

Then as the search of the cycle will be executed until n-1 = 0, then the cycle will be n * (n-1) which in the worst case the algorithm will be of order O (n^2).

4) Practice for midterms

4.1.1

		C	A	L	L	E
	0	1	2	3	4	5
C	1	0	1	2	3	4
A	2	1	0	1	2	3
S	3	2	1	1	2	3
A	4	3	2	2	2	3

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

4.1.2

		M	A	D	R	E
	0	1	2	3	4	5
M	1	0	1	2	3	4
A	2	1	0	1	2	3
M	3	2	1	1	2	3
A	4	3	2	2	2	3

4.2) 4.2.1 O(|x||y|).
4.2.2 return tabla[lenx][leny];

4.3) 4.3.1 O(n).
4.3.2 a) T(n) = c1:n + c2

4.4) O(n^2)

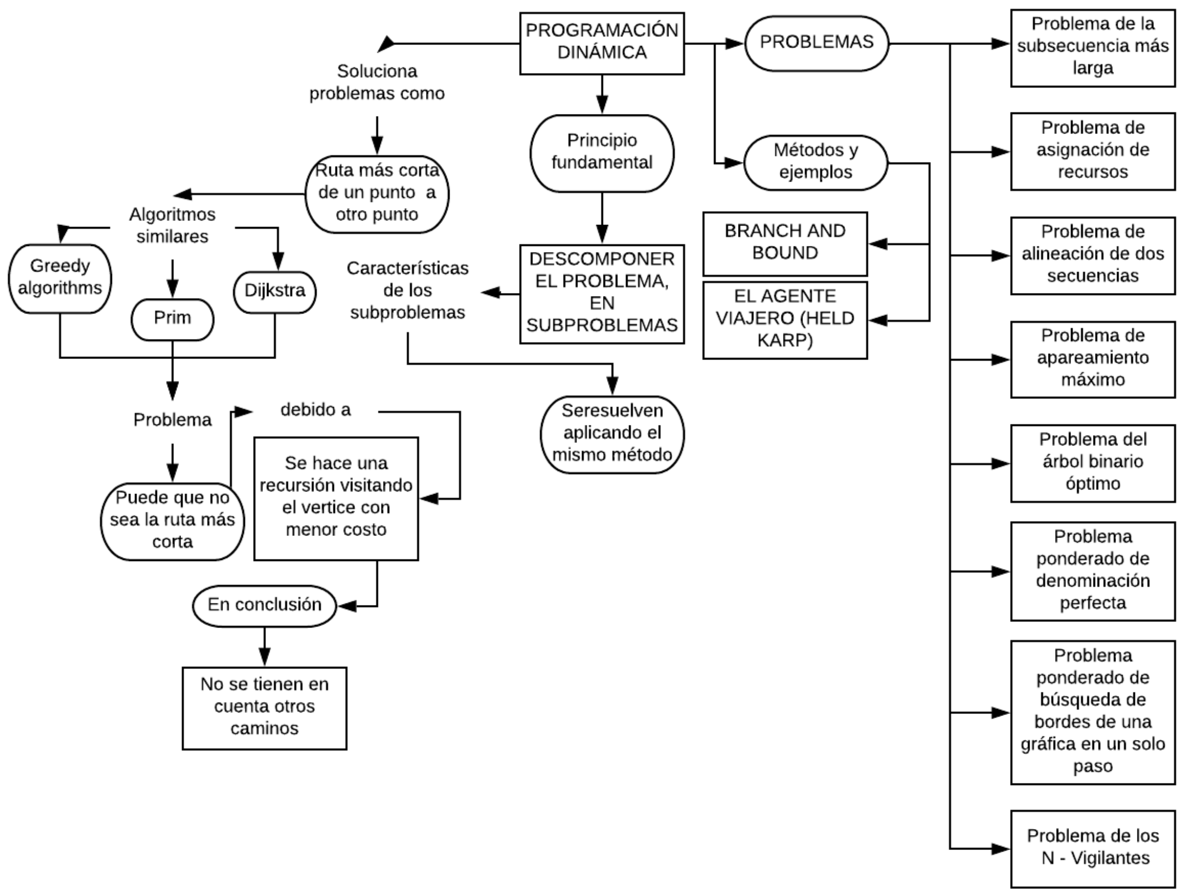
4.5) 4.5.1 c) T(n)=T(n/2)+C que es O(log n)
4.5.2 return a[mitad];
4.5.3 return bus(a, iz, mitad+1, z);

4.6) 4.6.1 scm[i];
4.6.2 scm[i] = scm[j] + 1;
4.6.3 max = scm[i];
4.6.4 d) O(nlogn)

4.7) 4.7.1 ni = d[i,j];
4.7.2 nj = d[k,j];
4.7.3 nk = d[i,k];
4.7.4 O(n^3)

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

5) Recommended reading (optional)



6) Team work and gradual progress (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