Laboratory practice No.5: Divide and conquer and Dynamic programming

María Sofía Uribe
Universidad Eafit
Medellín, Colombia
msuribec@eafit.edu.co

Isabel Cristina Graciano
Universidad Eafit
Medellín, Colombia
icgracianv@eafit.edu.co

3) Practice for final project defense presentation

3.1 We implemented the solution to the TSP problem (with dynamic programming) using an adjacency matrix to represent the graph. The algorithm generates the subset of each node to all other nodes (without the first vertex) ,then the distance from the initial node to the subset and the distance from this subset to all other subsets are computed and stored in the cost matrix . After these computations are performed the smallest distance is chosen.

3.2 The complexity of the algorithm is O(n^2 * 2^n) so the number of computations needed would be (50^2 * 2^50) which is 2.8147498e+18 .

3.3 The question is the same as the one answered in 3.4

3.4 We chose to represent the world of the robot as a graph (implemented as an adjacency matrix). The matrix holds the position of each pile of rubbish and the initial position.

The algorithm uses the past implementation of the held karp algorithm to find the cost of the shortest path

3.5 O(n^2 * 2^n)

3.6 n is the size of the matriz, that is the number of piles of rubbish + 1 (because of the initial position).

4) Practice for midterms

4.1 Levenshtein distance

4.1.1 (getting from casa to calle)

	ϵ	c	a	l	l	e
ϵ	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

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

a 4 3 2 2 2 3

4.1.2 *mama - madre*

		c	m	a	d	r	e
	c	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 Longest subsequence

4.2.1 *O(lenx*leny)*

4.4.2 `table[lenx][leny]`

4.3 nth element of Fibonacci

4.3.1 a) *O(n) , se hace n veces el ciclo*

4.3.2 a) *c1:n + c2*

4.4 improving Fibonacci

4.4.1 c) *O(2^n) y se optimiza con programación dinámica*

4.5

4.5.1 c) *T(n)=T(n/2)+C que es O(log n)*

4.5.2 *a[mitad]*

4.5.3 *a, mitad + 1, de, z*

4.6

4.6.1 `scm[i]=1;`

4.6.2 `scm[i]= 1+ scm[j]`

4.6.3 `max = scm[i];`

4.6.4 c) *O (n^2)*

4.7

4.7.1 `d[i][j];`

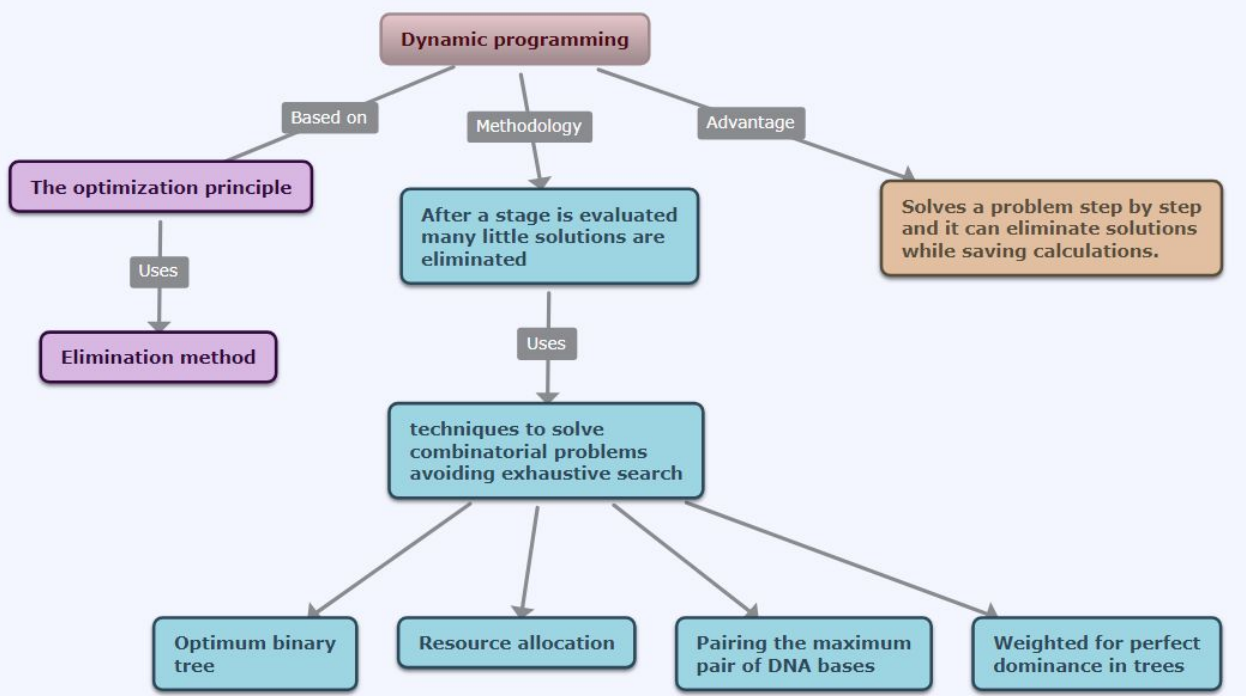
4.7.2 `d[k][j];`

4.7.3 `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) Discussing

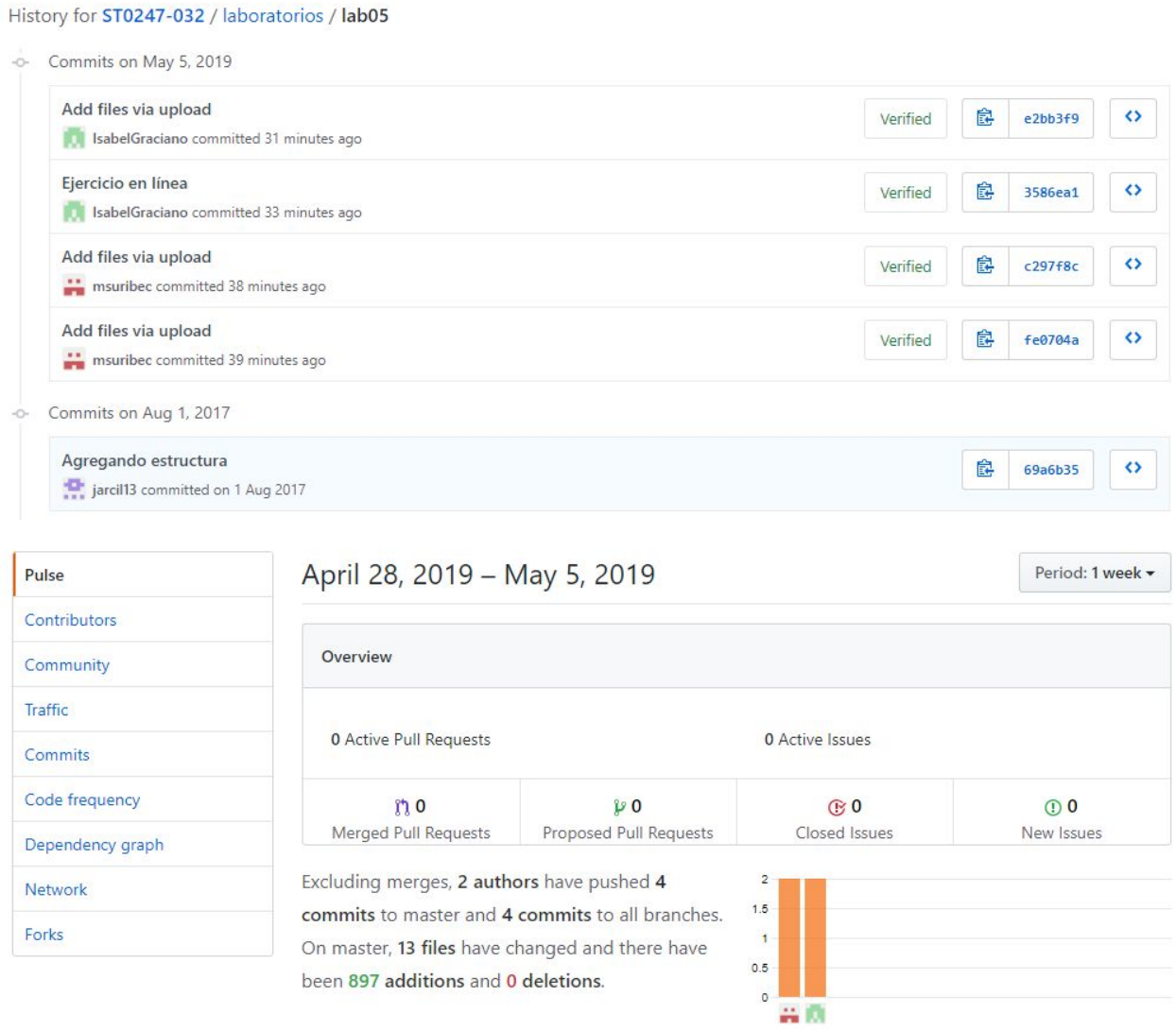
6.1 Meeting minutes

TEAM MEMBER	DATE	DONE	DOING	TO DO
ISABEL	02/05/2019	Discussing task distribution		Writing laboratory
SOFIA	02/05/2019	Discussing task distribution		Writing laboratory
SOFIA	03/05/2019	Exercise number 2		Code comments and practice for midterms
SOFIA	03/05/2019	Answer questions (practice for final project)		
ISABEL	04/05/2019	Concept map and optional reading		Practice for midterms
SOFIA	04/05/2019	Practice for midterms		
ISABEL	04/05/2019	Practice for midterms		

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

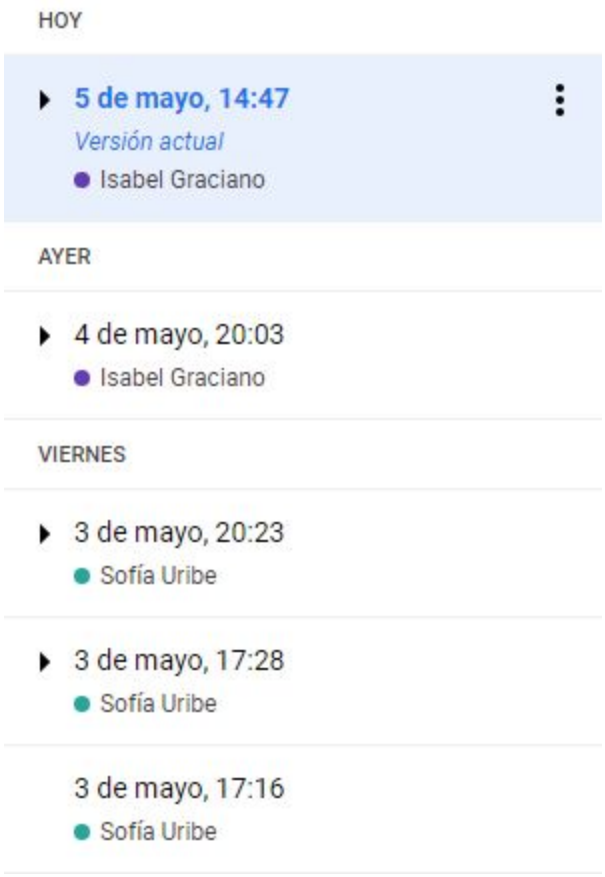
SOFIA	05/05/2019	Upload the laboratory report		
ISABEL	05/05/2019	Upload files .java		
SOFIA	05/05/2019	Upload files .java		

6.2Git report



6.3 Google doc history

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