Laboratory practice No. 1: Regression

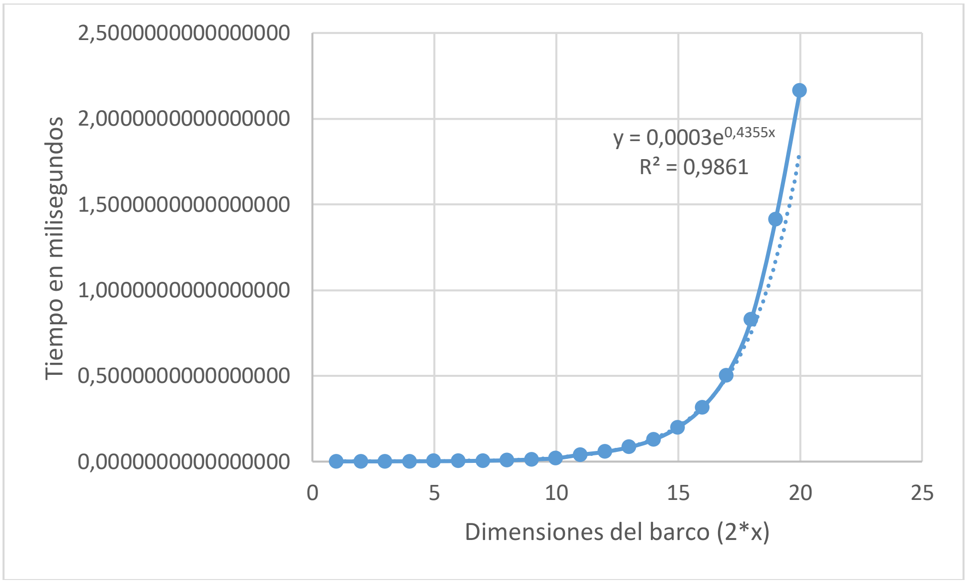
Santiago Cartagena Agudelo
Universidad Eafit
Medellín, Colombia
scartagena@eafit.edu.co

Paulina Pérez Garcés
Universidad Eafit
Medellín, Colombia
pperezg@eafit.edu.co

3) Practice for final project defense presentation

3.1 *T(n)= C + T(n-1) + T(n-2), or, expressed with Big O Notation, O(n)= O(2^n).*

3.2 *Once we graphed the times it takes for our algorithm to run using numbers from one to twenty, we were able to find (with a little help from Excel) an exponential function to calculate the time for different quantities. The graph is shown below, as well as the time it would take us to find the possible sorting of containers in a ship that is 2cmx50cm long.*



$$t(50) = 0{,}0003e^{0{,}4355(50)} = 858783.32\ ms = 858.78\ s = 15\ min$$

After running the algorithm for a structure with these dimensions we noticed that our previous guess had been off. In reality, during this test run. The real results read

$$t(50) = 3979240.97\ ms = \ 3979.24\ s = 1h6min$$

3.3 *This algorithm wouldn't be useful for Puerto Antioquia. Its implementation there would require calculating it for ships with different and larger dimensions and therefore it would take a huge amount of time when not used in a supercomputer. All that, in addition to the fact that each ship would carry at least 3 layers of containers, makes it non-viable.*

3.4 *GroupSum5 is an algorithm similar to the classic GroupSum. It searches for a way to add a certain combination of numbers given in an array and then returns true if it is possible. The exception here is that all multiples of five (numbers where n%5==0) must be considered when looking for the different possibilities and, if a number one follows one of these, then it must be skipped.*

For its implementation the method uses three parameters: start, nums and target. Start is the index of the array it will examine during a certain moment; nums is the array of numbers from which the addition must be made; finally, target is the number we're trying to reach by adding the ones in nums.

At first the algorithm works like GroupSum, checking, once start is on the last index of the array, if the target has been reached (target==0) and returning true if it has and false if it hasn't. Afterwards it checks if the next number is divisible by 5, has a following number (start<nums.length-1) and if it does, checks if it is a one; once all of that has been marked as truth we subtract the multiple of 5 and run the method with start being the number that goes after the one. Next condition is if the number at the start index if divisible by five, if it is it is considered for the sum. Finally, we run the method once again, checking the next index (start+1) and searching for solutions whether the number at start is taken into account or not.

3.5

EXCERCISE	LEVEL	COMPLEXITY (T(N))	COMPLEXITY (BIG O)
1. *Factorial*	1	$T(n) = C_1 + T(n-1)$	$O(n)$
2. *Fibonacci*	1	$T(p) = C_1 + T(p-1) + T(p-2)$	$O(2^p)$
3. *Triangle*	1	$T(q) = C_1 + T(q-1)$	$O(q)$
4. *sumDigits*	1	$T(r) = C_1 + T(r/10)$	$O(\log r)$
5. *countX*	1	$T(r) = C_1 + T(r-1)$	$O(r)$
1. *groupSum6*	2	$T(m) = C_1 + 2T(m-1)$	$O(2^m)$
2. *grouoNoAdj*	2	$T(m) = C_1 + 2T(m-1)$	$O(2^m)$
3. *groupSum5*	2	$T(m) = C_1 + 2T(m-1)$	$O(2^m)$
4. *groupSumClump*	2	$T(m) = C_1 + 2T(m-1)$	$O(2^m)$
5. *splitArrayHelper*	2	$T(m) = C_1 + 2T(m-1)$	$O(2^m)$

3.6

VARIABLE	DEFINITION
n	*Number for which the factorial must be found.*
m	*Difference between the length of the array and the index referenced with 'start'. (Usually equals the length of the array)*
p	*Position from the Fibonacci sequence to be found.*
q	*Rows in the triangle*

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

r	*Length of the array/string.*

4) Practice for midterms

4.1 *Start + 1, Nums, Target*
4.2 *a.*
4.3
- **4.3.1** *int res = solucionar(n-a,a,b,c) + 1;*
- **4.3.2** *res = Math.max(res, solucionar(n-b,a,b,c)+1));*
- **4.3.3** *res = Math.max(res, solucionar(n-c,a,b,c)+1));*

4.4 *e.*
4.5
- **4.5.1** *Linea 2: return n; Linea 3: return formas(n - 1) + , Linea 4: formas(n - 2);*
- **4.5.2** *b.*

4.6
- **4.6.1** *return (charAt(i-1) - 'o');*
- **4.6.2** *return (charAt(i) - 'o') + (charAt(i + 1) - 'o';*

4.7
- **4.7.1** *return comb(S, i + 1, t);*
- **4.7.2** *return comb(S, i + 1,t –S[i]);*

4.8
- **4.8.1** *return 0*
- **4.8.2** *int suma = ni + nj;*

4.9 *c.*
4.10 *b.*
4.11
- **4.11.1** *return lucas(n – 1) + lucas(n – 2)*
- **4.11.2** c.

4.12
- **4.12.1** return 0;
- **4.12.2** return sat += Math.max(Fi, Fj);
- **4.12.3** return sat;

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