Laboratory practice No. 1: Recursion

Juan Andrés Arroyave
Universidad Eafit
Medellín, Colombia
jaarroyavs@eafit.edu.co

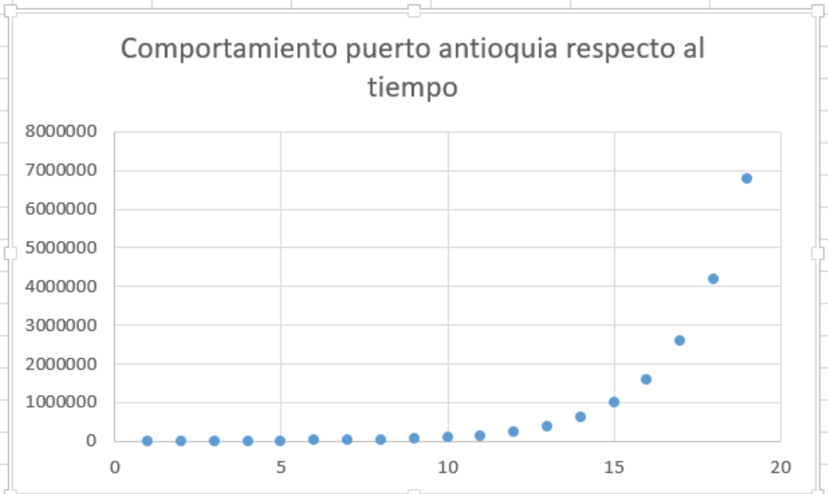
Gustavo López Garcia
Universidad Eafit
Medellín, Colombia
galopezg@eafit.edu.co

3) Practice of questions of sustentación of projects

***3.1** The complexity of this algorithm in notation O is a complexity of O (N ^ 2).*

3.2

1	0
2	1001
3	2001
4	4002
5	7003
6	12004
7	20010
8	33014
9	54021
10	88043
11	143121
12	232459
13	376359
14	609376
15	986869
16	1600559
17	2586432
18	4185953
19	6772276



The time it takes to calculate the time for 50x2Cm ^ 2 with rectangles of 1x2cm ^ 2 is 3.0414093201713378043612608166065e + 64.

***3.3** This algorithm is not suitable for use in the Antioquia port since its growth over time is too much, which is why it is better to use other solutions.*

***3.4** In the exercise groupSum5 we intend to return a boolean variable that says if in a group of numbers belonging to an array they can add a defined value (target) first we look if we find a value in the array that is a multiple of 5, we look at whether this it is fulfilled and we continue going through the arrangement until we reach our stop condition. If in the arrangement this is not fulfilled the stop condition will be that target is 0, since if this does not happen we will exhaust the stack and have a case of StackOverFlow. The description works in this way. (Solution):*

```
public boolean groupSum5(int start, int[] nums, int target) { if(start >= nums.length) return
target == 0; if(nums[start]%5==0){
target -= nums[start];

if(start+1<nums.length && nums[start+1]==1) start++; return groupSum5(start+1, nums, target);
}

return groupSum5(start+1, nums, target - nums[start])|| groupSum5(start+1, nums, target);
}
```

3.5 Complexity of CodingBat exercises:

Section one:

***Factorial:** This function grows at a monstrous pace, calculating its level of growth is approximately 2 ^ n, that is, it requires 2 ^ n steps to perform very large factorials.*

In Big O notation this is : O(2^n)

***BunnyEars:** This function maintains a constant growth along all the parameters that we pass, therefore the change ratio that is presented is a straight line. Therefore we can conclude that its growth is linear In notation Big O would be O (n).*

***NoX :** This exercise tries to eliminate all the x that appear in a string, it works in the following way:*

Basic step: if the string has a size of 0, so the empty string is returned // C1

Inductive step: If we see that in the first letter an x appears, we add one and continue going through the arrangement from the position i + 1-th

End of the recursion: After this, we add the first position if the x is not, since in the recursion or induction, the x have been eliminated and concatenated by making a recursion from the first position.

The complexity of said algorithm looks like this:
C1 if (str.length () == 0)

C2 + noX (str.substring (1)) // C2 + noX increasing one position if the condition is true.

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

C3 + noX (str.substring (1)) // Another constant plus an increasing position

T (n) = "" if the string is empty

NoX (str.substring (i + 1)) + C if the length of str is greather than 0.

Count8: ***The complexity of this algorithm is O (log (n))***

This is according to the rules that are defined in chapter 3 of the book Introduction to Algorithms 3rd edition.

PowerN: ***This algorithm has a complexity of 2n-1 which in O notation would be equivalent to O (2 ^ n-1).***

4) Practice for midterms

4.1 *groupSum (start+1, nums, target)*
4.2 *a*
4.3 Línea 4: n-a, a, b, c
Línea 5: res, solucionar(n-b, a, b, c)
Línea 6: res, solucionar (n-c, a, b, c)
4.4
4.5 Línea 2: return n;
Linea 3: return formas(n-1) + formas (n-2);
4.6 Linea 10 : SumaAux(n,i+2)
Linea 12 : SumaAux(i,n+1);
4.7
4.8 Return 0 ;
Linea 13: return ni+nj;
4.9
4.10 Correct Answer : 6.

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