Laboratory practice No. 01: Recursion

Catalina López Roldan
Universidad Eafit
Medellín, Colombia
clopezr9@eafit.edu.co

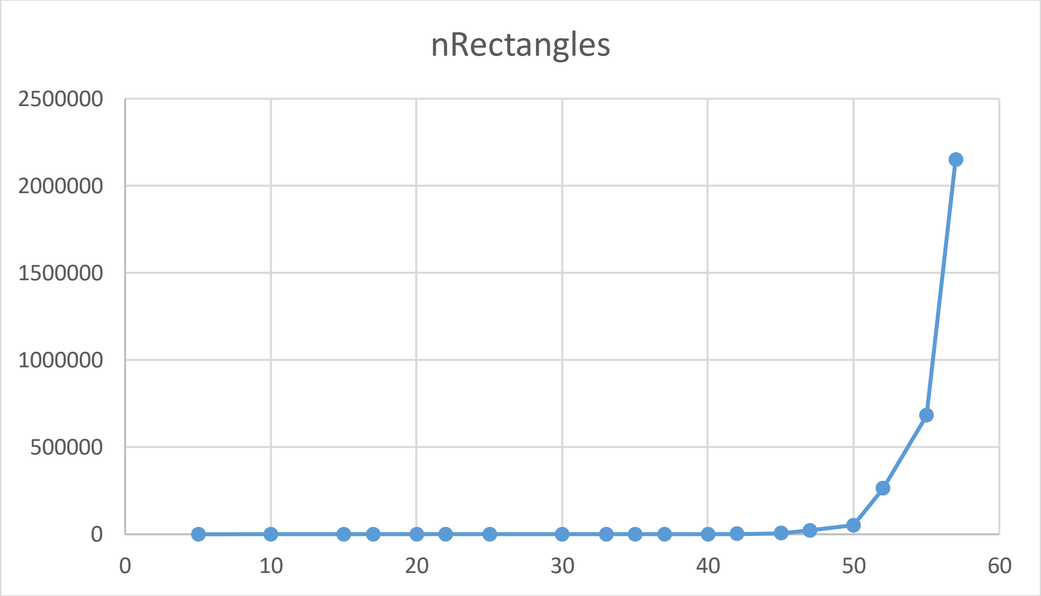
Andrés Darío Chaves Pérez
Universidad Eafit
Medellín, Colombia
adchavesp@eafit.edu.co

3) Practice for final project defense presentation

3.1 T(n)=T(n−1) + T(n−2) + C

3.2

Items	time(ms)
5	0
10	1
15	4
17	1
20	3
22	2
25	2
30	17
33	55
35	137
37	202
40	819
42	1623
45	4864
47	21641
50	51310
52	262328
55	681448
57	2149811



3.3 For us the use of these algorithm in 2020 is not viable, we can say it because his complexity is exponential, which means that it takes a lot of time to calculate the cipher of how many ways the port can be organized, and in a port that is going to be massive, receiving thousands of containers in a day these process needs to be fast.

3.4 The input parameters are an integer start that shows the start of that loop, because a recursive method can be also seen as a loop; an integer array nums, that are the numbers the code will check to subtract from the target that is another integer and is the achievement of the code by adding up specific numbers.
The code starts having three if statements, two of them have nested if statements. The first if is the stop condition of the recursive call and its nested if checks if the subtraction of a group of numbers always including the multiples of five makes the target zero, returning true as result, but if is not possible it returns false.
The second if, checks if the number of the array is divisible by five, if the condition is true makes a recursive call changing start parameter by adding one and subtracting the number in the actual start position; else enters into another if statement that checks if the next position of the array is a one so it does not take into account that five that is before that one, so it returns a recursive call adding two to the start parameter and subtracting the target the number of the array in the actual start position. The third if is for the numbers that are not divisible by five, so it makes another recursive call changing start parameter by adding one and subtracting the number in the actual start position and if the recursive returns true it also returns true. Finally, the method has a return statement that is call up when all the other statements are false making a recursive call giving as parameters start plus one, the array of numbers and the target.

3.5

Count7: T(n) = T(n/10) + k

CountX: T(n) = T(k)
BunnyEars2: T(n) = T(n - 1) + k
PowerN: T(n) = T(n - 1)
ChangePi: T(n) = T(2k)
GroupSum: T(n) = T(n^2) + k
sidesAreEqual53: T(n) = T(n + 1) + k
sidesAreOdd10: T(n) = T(n + 1) + k
sidesAreEqual: T(n) = T(n + 1) + k
groupNoAdj: T(n) = T(n + 1) + k

3.6

k= constant
n= condition that change every time that a recusive call is made

4) Practice for midterms

4.1
4.2 B
4.3
4.3.1 n-a, a , b, c
4.3.2 res, solucionar(n-b, a, b, c)
4.3.3 res, solucionar(n-c, a, b, c)
4.4 [Optional]
4.5
4.5.1 line 2 : return 0;
line 3 : n-1
line 4 : n -2
4.5.2 T(n) = T(n−1) + T(n−2) + C
4.6
4.6.1 Line 10: sumaAux(n, i + 1)
Line 12: sumaAux(n, i + 1)
4.7
4.8
4.8.1 line 9: return false;
line 13: nj || cuantas(K, v, n - 1);

4.9 C = 22
4.10 B = 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