Lab Nro. 1: Recursion

Sebastian Loaiza Correa
Universidad Eafit
Medellín, Colombia
sloaizac@eafit.edu.co

Cristian Dario Ceballos Rodriguez
Universidad Eafit
Medellín, Colombia
cdceballor@eafit.edu.co

2)

2.3 GroupSum5: It is an algorithm that through an arrangement tries to "separate" the numbers that are multiples of 5 and add them, although there are some restrictions, such as if a number n which is multiple of 5, and the next one, that is n + 1, results in a 1, this last digit can not.
Now, the algorithm works first by finding a number that is a multiple of 5 in the array (n% 5 = a 0) and if this happens then it evaluates if the integer total is equal to the number of positions in the array and search among them those that are multiples of 5 which accumulates them in a final variable which is sum through recursion.
2.4
Factorial: T (n(n-1))
BunnyEars: T(n(n+2))
PowerN: T(c^(n))
Array6: T (n+1)
Triangle: T (c + n (c2-1)

groupSum5: T(n(2n/5) +c1+c2)
groupNoAdj: T(n(2n) c1+c2)
groupSumClum: T(n(2n+n/2+c1-c2))
sidesAreEqual: T(n*n+c1-c2)
sidesAreOdd10: T(n(c1+c2) n*n)
2.5 In recursion exercises 1, we usually work with a single constant, unlike recursion 2, where we work with 2

3)Mock project support questions

3.1 Stack Overflow: It is an error that tries to overload the battery, it is when the battery overflows with data that ARE extremely large or a very large amount of data which exceeds the capacity of memory of the computer destined for Stack.
3.2 The number 45 of the series was the largest value that we could calculate with the algorithm; It can not calculate Fibonacci with a million since when accumulating so much data in pile the process would collapse or it would be very extensive execution time.
3.3 Optimizing the algorithm to improve its execution time and allow the series to be calculated up to a very large number n

4)Simulacro de Parcial

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

1. start + 1, nums, target
2. c
3. 3.1 n-a, a, b, c
   3.2 res, solucionar (n-b, a, b, c) +1
   3.3 res, solucionar (n-c, a, b, c) +1
4. *e*
5. 5.1 2. return n
   3. n-1
   4. n-2
   5.2 b
6. 6.1 sumaAux (n, i+2)
   6.2 sumaAux (n, i+1)
7. 7.1 return comb (S, i+1, t - S[i])
   7.2 return comb (S, i+1, t-1)

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co