

Laboratorio Nro. 1: RECURSIÓN

Santiago Hincapié Murillo
Universidad Eafit
Medellín, Colombia
shincapiem@eafit.edu.co

Santiago Valencia Arango
Universidad Eafit
Medellín, Colombia
svalenciaa@eafit.edu.co

Andres
Almanzar Restrepo
Universidad Eafit
Medellin, Colombia
aalmanzarr@eafit.edu.co

2) Ejercicios 2.3 , 2.4, 2.5

2.3) Primero da una condición de parada, luego comprueba si el número en la posición [start] es múltiplo de 5 o no, si lo es lo añade a la suma. Luego de esto suma los siguientes número del arreglo pudiendo tomarlos o no para llegar a Target

Codigo tomado de gregorulm.com Codignbat:Java. Recursion2 Marzo 23 2013

2.4) Fibonacci:
 $O(c^n)$

PowerN:
 $O(c^n)$

BunnyEars:
 $O(n)$

Factorial:
 $O(n!)$

Triangle:
 $O(n)$

2.5) Las n, m etc, Representan la cantidad de veces que un algoritmo ejecuta una instruccion.

3) Simulacro de preguntas de sustentación de Proyectos

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

1. Stack Overflow o desbordamiento de la pila, es un error que puede ocurrir con frecuencia cuando se sobrepasa la capacidad que posee la variable que se ha asignado. Aquel error o excepción que mandan los lenguajes: En java o como se muestra en ruby (SystemStackError).
2. El número en el tiempo de ejecución era demasiado lento fue 55, con este número se registró un tiempo de ejecución de 9 min y 27 s. No es posible hacer el Fibonacci de 1 millón debido a que salta el StackOverflow.

3. se puede usar la fórmula de Binet.

Desde 1841, sabemos que el término n -ésimo de la sucesión de Fibonacci, puede calcularse por la llamada fórmula de Binet (Jacques Binet, 1786 - 1856). Aunque, en realidad el primero en demostrarla fue Abraham de Moivre (1667 - 1754).

información tomada de : <https://www.ugr.es/~eaznar/fibo.htm>

4. La complejidad de recursion2 (2.4) es mayor a la de recursion1 pues estos un solo llamado recursivo lo que permite operar valores más grandes. Mientras que los de recursion2 hacen al menos 2 llamados lo que hace que tome más tiempo pues como mínimo son de complejidad $O(2^n)$

4) Simulacro de Parcial respuestas:

1. $Start + 1, nums, target$
2. $D \cdot T(n) = T(n - 1) + C$
3. 3.1) n, a, b, c
3.2) n, res
3.3) n, res
4. ¿Qué calcula el algoritmo desconocido y cual es su complejidad asintótica en el peor de los casos?
d. la suma de los elementos de arreglo a y es $O(n)$
5. 5.1
Línea 2: Return 1;
Línea 3 : $n - 2$;
Línea 4: $n+1$;

5.2
d. $T(n) = T(n+1)+C$
6. 6.1
Línea 10: $0 + suma(5,i+1)$;
6.2

Línea 12: suma(5,i+1);

7. S, I, T