

## Laboratorio Nro. 1: Recursión

**Andrés Sánchez Gómez**  
Universidad Eafit  
Medellín, Colombia  
asanchezg@eafit.edu.co

**David Calle Daza**  
Universidad Eafit  
Medellín, Colombia  
dcalled1@eafit.edu.co

2)

**2.3** Éste ejercicio consiste en hallar la manera de que el algoritmo decida si es posible hallar la suma de un número determinado contando todos los múltiplos de cinco que se hallen y omitiendo los unos que los siguen. La solución de éste problema empieza por revisar el finl del arreglo, ya que así podemos revisar si sí se alcanzó la suma deseada, en ese entonces se retorna verdadero o falso dependiendo del caso. Después tenemos que revisar si el número actual es un múltiplo de 5, si en efecto, es múltiplo de 5, se procede a revisar si el número siguiente es 1 y si así es se ignora, luego se busca la posibilidad de hacer la suma deseada restándole el múltiplo de cinco con los demás elementos del arreglo. En caso de que el número no sea un número de 5, se revisa si la suma es posible incluyendo y no incluyendo el número actual. Por último, si no se retorna ningún verdadero, entonces se retorna falso para mostrar que no es posible.

**2.5** 'n' y 'm' son las variables que son ingresadas a los métodos, para que de esta manera puedan hacerse de manera adecuada y teniendo una idea de lo largo que son estos mismos. Las variables anteriores representan la longitud del problema que se le va a ingresar a cada método, afectando de manera directa la complejidad del mismo.

3)

**3.1** Stack overflow es una plataforma que nos permite encontrar soluciones a problemas de problemas de programación en distintos lenguajes. Fue de gran ayuda al momento de solucionar algunos ejercicios. Aprendimos que esta plataforma puede ser de gran ayuda al momento de no hallar alguna solución óptima para nuestros programas o bien sea que necesitemos de una mejora de la misma.

**3.2** 70, mas no por error de Stack Overflow, si no porque al ingresarle éste valor tardó en ejecutarse demasiado tiempo, por lo que era poco efectivo intentarlo con arreglos de mayor tamaño, puesto que sería imposible graficarlo. Por éste motive y por el de stack overflow es que no podríamos ejecutar Fibonacci con 1 millón.

**3.3** Para calcular valores grandes con Fibonacci podemos hacer un proceso de memorización en la memoria, el cuál consiste en guardar los valores de Fibonacci requeridos para la realización de otros. De está manera

no tendríamos que hacer le cálculo completo con algunos valores, ya que podemos recuperarlos directamente de la memoria

**3.4** Después de realizar los 5 ejercicios de recursion 1 y los 5 de recursion 2 notamos que en recursion 1 tratamos con algoritmos más sencillos, en los que no tenemos que implementar arreglos, mientras que en recursion 2 encontramos que todos los problemas son con operaciones de arreglos.

#### **4) Simulacro de Parcial**

1. *Start + 1, nums, target*
2. *c*
3. *length-1*
4. *e*
5. *return n, n-1, n-2 y b*
6. *sumaAux(n, i+2) y sumaAux(n, i+1)*
7. *(S, i+1, t-s[i]) y (S, i+1, t)*