David Marquez Mínguez

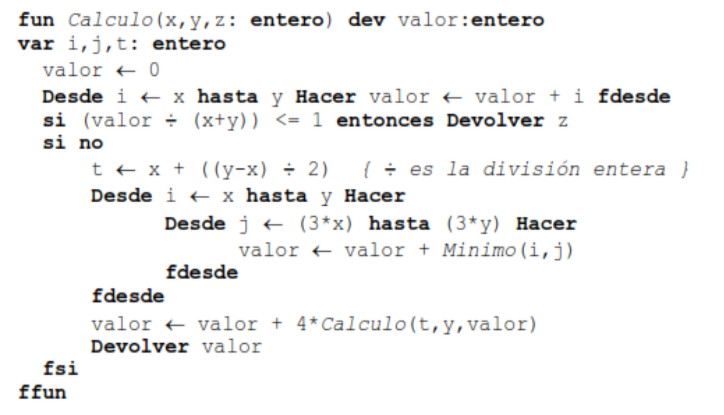
47319570Z

Ejercicios TEMA 1

ALGORITMIA Y COMPLEJIDAD

# EJERCICIO 3

El ejercicio, pedía realizar un análisis de un determinado fragmento de código el cual en concreto era el siguiente.



Aplicando la teoría del calculo de la eficiencia, calculamos la ecuación recursiva de la misma y la resolvemos. A continuación, se muestra una imagen con la ecuación y la resolución de esta.

Imagen que contiene texto

Descripción generada automáticamente

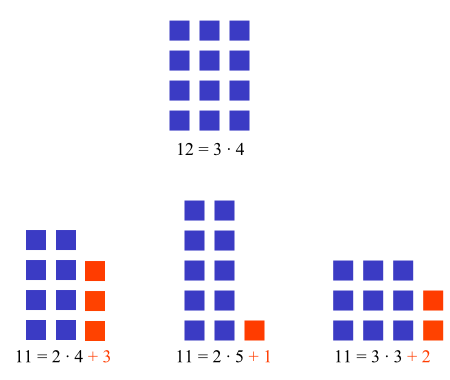
Como podemos observar, la ecuación presenta una complejidad de n^2.

# EJERCICIO 7

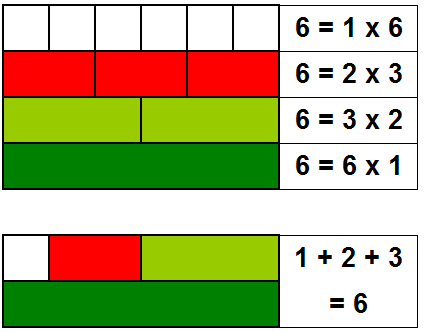
En este ejercicio, se pedía que el usuario introdujese un número, y a partir del mismo determinar cuantos números primos y cuantos números perfectos había entre 1 y ese número.

Para entender primero el problema, debemos conocer que es un numero primer y que es un numero perfecto.

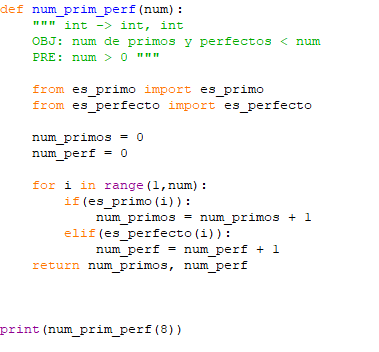
Un numero primo es aquel que tiene como divisores el 1, y el mismo, es decir, que si queremos dividir ese numero y obtener un numero entero, lo debemos dividir o por el 1 o por él mismo. Por ejemplo, el 11 es un numero primo, pero el 12 no lo es.



Por otro lado, un número perfecto es aquel el cual la suma de sus divisores produce como resultado el mismo número. Es decir, el 6 es un numero perfecto pues sus divisores son 1, 2, 3, si hacemos su suma, 1 + 2 + 3 = 6.



Una vez que se ha entendido que es un numero primo y que es un numero perfecto, se procede a la implementación del algoritmo. Tanto ahora, como en el resto de las prácticas, se empleará como lenguaje de desarrollo Python. A continuación, se muestra el código y un ejemplo de su salida por pantalla, así como una breve explicación de su funcionamiento.



Al principio del programa se incluye la importación de dos funciones auxiliares programadas por mis, las cuales serán necesarias para determinar si en primer lugar un número es primo, y, en segundo lugar, si un número es perfecto. A partir de ahora lo único que hay que g=hacer es ir recorriendo los números desde 1 hasta N, e ir comprobando de uno en uno si es perfecto o primo. Cuando se cumpla alguna de las condiciones se suma un valor al contador correspondiente. Finalmente, el programa devuelve el numero de primos y perfectos encontrados respectivamente. A continuación, se muestra su salida por pantalla.



# EJERCICIO 9

Se pide realizar una función recursiva que calcule el sumatorio desde 1 hasta un determinado numero indicado por el usuario. Es decir:

En cuanto a la implementación, es bastante simple. Lo único que hay que hacer es sumar valores desde N e ir restándole 1. La condición de parada es que dicho numero llegue a 1. S e muestra la implementación a continuación, así como su salida por pantalla.

