TRABAJO DE ANÁLISIS DE ALGORITMOS

PRESENTADO A:

ANDRES FERNANDO VELASCO

PRESENTADO POR:

JEISON ANDRES FUENTES ORTEGA

230172007

JOHAN SEBASTIAN FUENTES ORTEGA

230172002

UNIDAD CENTRAL DEL VALLE

INGENIERIA DE SISTEMAS

SEMESTRE VII

2020

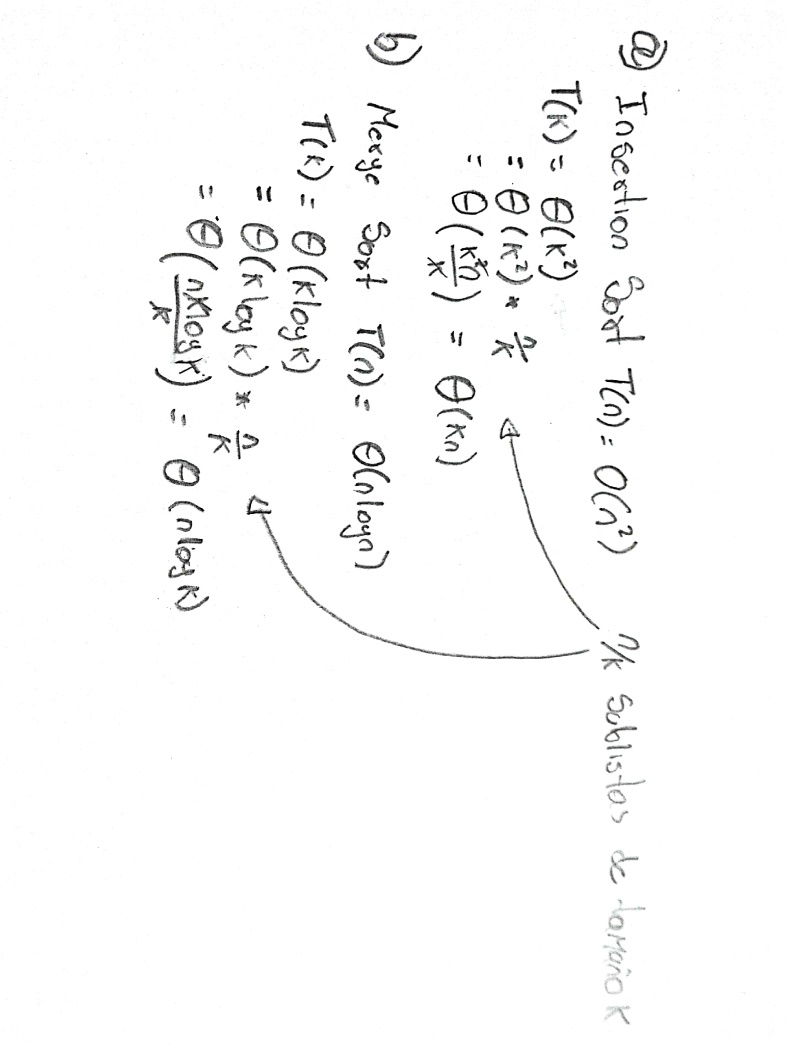
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 seg | 1 min | 1 hora | 1 día | 1 mes | 1 año | 1 siglo |
| Lg n |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| n |  |  |  |  |  |  |  |
| n lg n | 62725 | 2800000 |  |  |  |  |  |
|  | 1000 | 7745,9 |  |  |  |  |  |
|  | 100 | 391,4 | 1532,6 | 4420,8 | 13736,5 | 31593,8 | 146645,5 |
|  | 19,93 | 25,83 | 31,74 | 36,32 | 41,20 | 44,84 | 51,48 |
| n! | 9,4 | 11,1 | 12,7 | 13,99 | 15,2 | 16,13 | 17,75 |

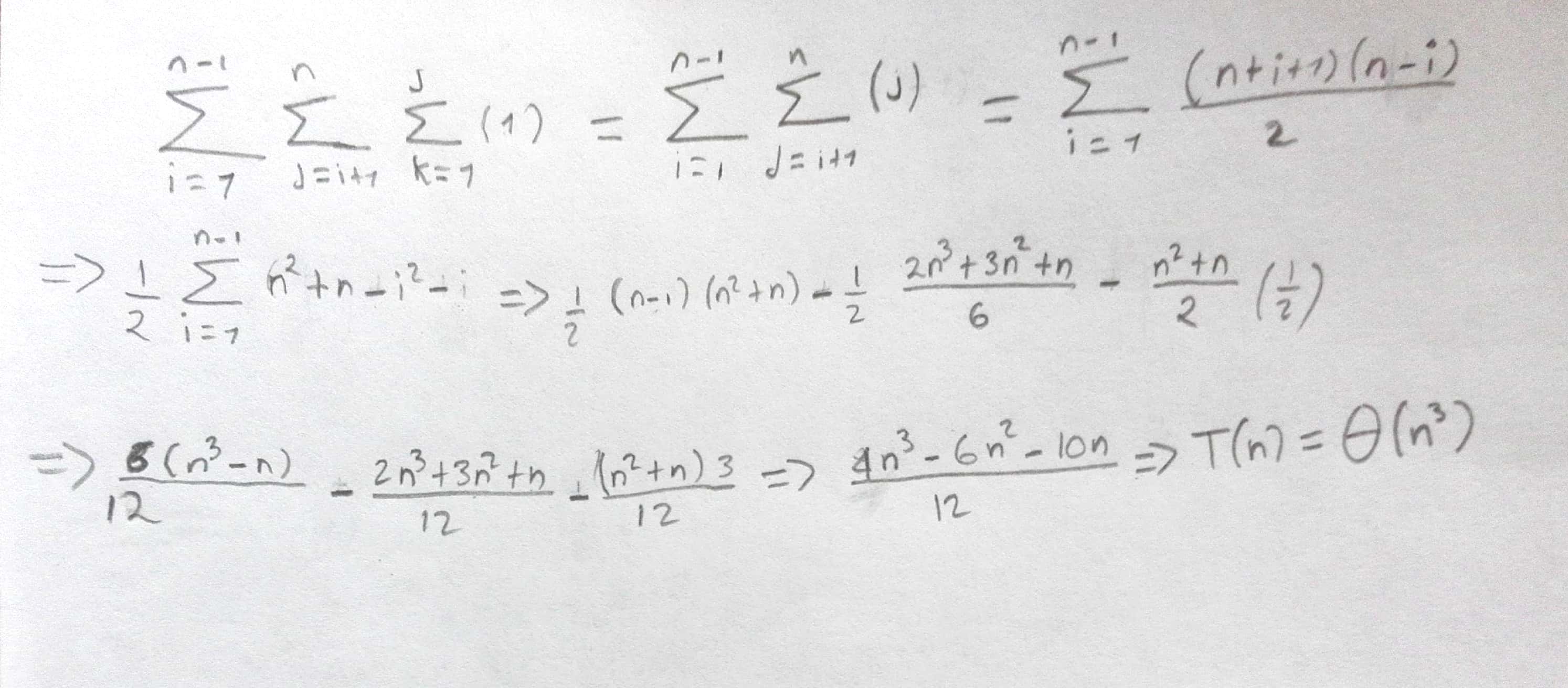
2)

Como dice que dado un conjunto S de números enteros y cuyo tamaño es n debemos determinar si existen 2 elementos cuya suma sea igual a un valor x entregado.

Ahora bien, para poder que el algoritmo tenga un comportamiento de (n lg n) , tenemos que hacer que entre todo el algoritmo, el peor caso sea (n lg n).

Entonces como primera medida lo mejor es realizar un ordenamiento al conjunto S, obviamente asignando S a una variable de tipo array, ese ordenamiento podemos realizarlo con el algoritmo de Merge sort, cuyo comportamiento es (n lg n). Una vez ordenado debemos recorrer el arreglo, tomamos el primer valor y restarle el valor de x, el resultado lo comparamos con el valor que se encuentra en la mitad del arreglo y determinar si es mayor solo tomamos de la mitad hacia el primer valor, de lo contrario tomamos de la mitad hacia el valor final del arregla, y así sucesivamente, básicamente aplicamos una búsqueda binaria, la cual implica un comportamiento lg n pero como se está recorriendo un ciclo de tamaño n, termina siendo un comportamiento n lg n. De esta manera, al aplicar solamente los algoritmos Merge sort y búsqueda binaria tendríamos que en el peor de los casos, el algoritmo tendrá un comportamiento (n lg n), implicando que cumplimos con lo que se nos ha pedido.





1. Sabemos que A es un arreglo de tamaño n que contiene las representaciones binarias de los números de 0 hasta n, excepto uno.

Se nos dice que para determinar el número perdido del array A, creamos un array auxiliar B que contiene todos los números de 0 a n; además se agrega que solamente se puede ir tomando el bit j-ésimo de A[i], el cual tendrá un comportamiento constante.

Básicamente hay 2 ciclos for, uno determinado de 0 a n y el otro determinado por el valor binario del número entero.

