

# Análisis de Algoritmos e introducción a Matemáticas Discretas

## Tarea 3

Rubén Pérez Palacios Lic. Computación Matemática  
Profesor: Dr. Carlos Segura González

15 de enero de 2024

La solución propuesta al problema es usar la técnica *divideandconquer* para encontrar la cantidad de sub-arreglos con suma igual a  $T$ , donde se divide por la mitad el arreglo y se resuelve recursivamente para cada mitad. Luego calculamos la cantidad de sub-arreglos que contengan al menos un elemento de cada mitad, como todo sub arreglo de este forma es también la unión de un sufijo y prefijo de la primera y segunda mitad respectivamente, entonces, realizamos un bucket sobre todas las posibles sumas de los sufijos de la primera mitad y después iteramos sobre la sumas de todos los sufijos y sumamos al total la cantidad de prefijos cuya suma es  $T - \text{suffix}$  donde *suffix* es la suma del sufijo actual. Para combinar las respuestas solo hace falta sumar las respuestas de las dos mitades y la respuesta descrita anteriormente.

La complejidad del algoritmo descrito es  $O(N \log^2(N))$  ya que la complejidad del trabajo hecho en cada nodo de recursión es de  $O(N \log N)$ . Esto también se puede ver en el árbol de recursión puesto que todos los trabajos tendrán este factor  $\log N_i$  donde  $N_i \leq N$  entonces podemos factorizarlo y seguir la misma demostración de la complejidad del *Mergesort* (o Master Theorem).