Análisis de Algoritmos y Matemáticas Discretas
Tarea 3 - Divide y Vencerás

Antonio Barragan Romero

8 de septiembre de 2023

Analisis de complejidad de : ***Sumas Parciales - Divide y Vencerás***

Solución:

Sea $T(n)$ la función que mide el tiempo de ejecución de `subarrays_sum_less_than` Podemo notar que Dividir el problem solo es calcular el punto medio del arreglo, lo cual se hace en tiempo constante. Luego de Manera recursiva resolvemos dos problemas de tamaño $n/2$, lo cual contribuye en $2T(n/2)$ al tiempo de ejecución. Analicemos pues el tiempo de ejecución de `merge`. Lo primero que hace es crear un `ordered_set` y recorriendo de la mitad del arreglo al incio agrega las sumas parciales (de izquierda a derecha). Vimos en clase que insertar es $O(\log(n))$ y como recorremos la mitad, pues hacemos $n/2$ inserciones, entonces el tiempo de ejecución de llenar el `ordered_set` es $O(n\log(n))$. Despues se recorre de la mitad al final del arreglo calculando sus sumas parciales, por cada suma r, buscamos en el `ordered_set` el numero de subarrays cuya suma es menor a $t - r$, para ello llamamos a `order_of_key`, por propiedades del `order_of_key` esto se hace en $O(\log(n))$ y como hacemos $n/2$ busquedas tenemos un tiempo de ejecución de $O(n\log(n))$.

Lo anterior nos dice que la complejidad del `merge` es $O(n\log(n))$. Se sigue que

$$T(n) = \begin{cases} k & \text{para } n \leq 2, \\ 2T(n/2) + O(n\log(n)) & \text{para } n > 2. \end{cases}$$

Se puede ver que lo anterior no entra en ningun caso del Teorema Maestro, por lo cual calculemos $T(n)$ y para facilitar los calculos supongamos que $n = 2^p$, entonces tenemos que $\log(n) = p$, por lo cual

$$\begin{aligned}
T(n) = 2T(2^p) &\leq 2T(2^{p-1}) + c2^p p \\
&\leq 2\left[2T(2^{p-2}) + c2^{p-1}(p-1)\right] + c2^p p \\
&= 2^2 T(2^{p-2}) + c2^p((p-1)+p) \\
&\leq 2^2\left[2T(2^{p-3}) + c2^{p-2}(p-2)\right] + +c2^p((p-1)+p) \\
&= 2^3 T(2^{p-3}) + c2^p\left((p-2)+(p-1)+p\right) \\
&\vdots \\
&\leq 2^p T(1) + c2^p \sum_{i=1}^{p} i \\
&= nk + cn\frac{\log(n)(\log(n)+1)}{2} \\
&= nk + cn\frac{\log(n)^2}{2} + cn\frac{\log(n)}{2} \\
&= O(n\log^2(n)).
\end{aligned}$$

De lo anterior tenemos que $T(n) = O(n\log^2(n))$.