

Ayudantía 2 - Solución ejercicio propuesto [I1 2021-1]

Mejor caso

Tenemos 3 partes en el algoritmo, las cuales son `partition`, `median_modificado` y `quicksort_modificado`, cada uno de estos llamando al anterior. Revisamos la complejidad de cada parte:

- a) `partition`: Siempre recorre todos los elementos de un array al que ordena. Su complejidad por ende es simplemente $\mathcal{O}(n)$.
- b) `median_modificado`: En el mejor caso, sabemos que encontrará en el primer intento a la mediana del arreglo que recibe como parámetro. Esto Tiene complejidad $\mathcal{O}(1)$. Ya que `median_modificado` hace las llamadas a `partition`, hasta el momento la complejidad es $\mathcal{O}(n)$.
- c) `quicksort_modificado`: La complejidad de `median_modificado` no es mayor al `partition` que llamaríamos en un `quicksort` común y corriente. Entonces tenemos para `quicksort_modificado` una complejidad $\mathcal{O}(n)$ en el mejor caso.

Peor caso

Revisando cada parte del algoritmo nuevamente:

- a) `partition`: En cualquier caso sabemos que `partition` recorre todos los elementos \implies complejidad $\mathcal{O}(n)$.
- b) En el peor de los casos, `median_modificado` sacará como pivote todos los elementos del arreglo antes de encontrar la mediana (es decir, sacará un pivote, verificará con `partition` que su índice corresponda a $\frac{n}{2}$, y quedarán $n - 1$ elementos a un lado, por lo que aplicará `partition` nuevamente sobre los $n - 1$ elementos). La complejidad es $\mathcal{O}(n^2)$ porque son n llamados a `partition`.
- c) Escribiendo la ecuación de recurrencia:

$$T(n) = \begin{cases} 1 & n = 1 \\ n^2 + T(\frac{n-1}{2}) & n > 1 \end{cases}$$

El término n^2 por la llamada a `median_modificado` y el término $2T(\frac{n-1}{2})$ porque se procede llamar a `quicksort_modificado` sobre 2 arrays de largo $\frac{n-1}{2}$.

Si se desarrolla $T(n)$, se llega a una suma de términos de n^2 , de modo que se obtiene $\mathcal{O}(n^2)$ como complejidad en el peor caso.

Comentarios respecto al algoritmo

Se puede argumentar que el compañero no mejora `quicksort` ya que se obtiene la misma complejidad en el peor caso y en el mejor.