

Pregunta 1

Comenzamos encontrando los valores de A, B y $f(n)$ según Teorema Maestro:

- $A = 4$, dado que se realizan 4 llamadas recursivas a algoritmo().
- $B = 2$, dado que cada llamada se realiza con un subArray de tamaño $n/2$
- $f(n) = n^2$, dado que el trabajo realizado fuera de las llamadas recursivas se realiza $n \cdot n$ veces. De esta forma, $d=2$

Tereoma Maestro:

De esta forma, la expresión queda dada por: $T(n) = 4T(\frac{n}{2}) + n^2$ Luego, calculamos: $n^{\log_{\{b\}}(A)} = n^{\log_{\{2\}}(4)} = n^2$ De esta forma, el teorema maestro nos dice que si $d = \log_{\{b\}}(A)$, la complejidad del algoritmo está dada por: $T(n) = O(n^d \log(n)) = O(n^2 \log(n))$

Pregunta 2

El algoritmo recibe un arreglo desordenado y sus índices de inicio, medio y final y retorna un arreglo ordenado. Lo hace de la siguiente forma:

- Si hay un solo elemnto, retorna el arreglo
- Divide el arreglo recursivamente en dos partes hasta que no pueda ser dividido
- Une los subarreglos más pequeños en nuevos arreglos ordenados

Ejemplo

Ejemplo, recibimos el arreglo `array = [9,6,2,9,1]`

- Se crean dos arreglos mas pequeños: `array1 = [9,6]` y `array2 = [2,9,1]`
 - `array1` se vuelve a dividir en `array11 = [9]` y `array12 = [6]`
 - `array2` se vuelve a dividir en `array21 = [2,9]` y `array22 = [1]`
 - `array22` se vuelve a dividir en `array221 = [2]` y `array222 = [9]`
 - `array11` y `array12` se unen ordenados en `array01 = [6,9]`
 - `array221`, `array222` y `array22` se unen ordenados en `array02 = [1,2,6]` Finalmente, `array01` y `array02` se unen ordenados en `arrayFinal = [1,2,6,9,9]`

Cálculo de complejidad

- Definimos A como la cantidad de subarreglos que se generan despues de cada división, por lo que $A=2$
- Definimos B como el tamaño de cada subarreglo. Si se divide por la mitad, $B=2$
- $f(n)$ es $O(n)$, dado que todas las operaciones se realizan para tamaño n. De esta forma, $d=1$

La fórmula del teorema maestro queda definida por:

$$T(n) = 2T(\frac{n}{2}) = O(n)$$

Luego, calculamos:

$$n^{\log_b(A)} = n^{\log_2(2)} = n^1$$

Como $d = \log_b(A)$, se trata del primer caso de Teorema Maestro y la complejidad del algoritmo está dada por:

$$T(n) = O(n^d \log(n)) = O(n \log(n))$$

Pregunta 3

El mejor caso se produce cuando el vector X está ordenado y solo debe realizar n iteraciones:

- Mejor caso = $O(n)$

El peor caso se produce cuando el elemento menor está al final del vector X , por lo que se deben realizar $n \cdot n$ iteraciones:

- Peor caso = $O(n^2)$