

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 4

Juan Navarro

Problema 1

Describa y analice un algoritmo que dado un arreglo de n números enteros (positivos o negativos), permita encontrar el subarreglo contiguo de suma máxima.

Idea de solución La solución se basa en que todo subarreglo está contenido completamente en una mitad del arreglo, o bien es la unión de un subarreglo del final de la primera mitad, con un subarreglo desde el comienzo de la segunda mitad.

El algoritmo corresponde a:

1. Obtener recursivamente el subarreglo de suma máxima de cada mitad
2. Buscar los subarreglos de suma máxima desde el centro hacia los extremos, y generar el subarreglo formado por la unión de ambos
3. Comparar los tres subarreglos obtenidos, y retornar el de suma máxima

El tiempo del algoritmo está dado por:

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n & n > 1 \end{cases}$$

que pertenece a $\Theta(n \log n)$.

Problema 2

Describa y analice un algoritmo que determine la mediana de una lista desordenada, en tiempo $O(n)$ en el peor caso.

Idea de solución El algoritmo es:

1. Dividir la lista en bloques de largo 5.
2. Obtener la mediana de cada uno, por ejemplo, usando insertion sort.
3. Obtener recursivamente la mediana de las $\lceil n/5 \rceil$ medianas anteriores.
4. Particionar el arreglo, usando ese valor como pivote.
5. Buscar recursivamente en la lista que corresponda de acuerdo al largo de las partes generadas por partition.

La cantidad de comparaciones está dada por:

$$T(n) = \begin{cases} O(1) & n \leq K \\ 6 \cdot \lceil n/5 \rceil + T(\lceil n/5 \rceil) + (n-1) + T(7n/10+6) & n \geq K \end{cases}$$

Donde $6 \cdot \lceil n/5 \rceil$ corresponde a encontrar las medianas de los subarreglos, usando que se puede encontrar la mediana de 5 elementos con 6 comparaciones, $T(\lceil n/5 \rceil)$ en encontrar la mediana de las medianas, $n-1$ por partition, y finalmente $T(7n/10+6)$ por la última llamada recursiva. Para el último caso, se usa que la mediana de las medianas es mayor o igual a al menos tres elementos de la mitad de los bloques menos 2: el bloque que contiene la mediana, y el último que puede tener menos elementos.

Luego, por inducción constructiva, buscamos demostrar que $T(n) \leq \kappa n$, para n suficientemente grande:

$$\begin{aligned} T(n) &\leq 6 \cdot \lceil n/5 \rceil + \kappa(\lceil n/5 \rceil) + (n-1) + \kappa(7n/10+6) \\ &\leq 6n/5 + 6 + \kappa n/5 + \kappa + (n-1) + \kappa 7n/10 + 6\kappa \\ &= kn + (-kn/10 + 7\kappa + 6n/5 + 5 + n) \\ &\leq kn \end{aligned}$$

La última desigualdad se cumple si $(-kn/10 + 7\kappa + 6n/5 + 5 + n) \leq 0$, o equivalentemente, si $n > 70$ y

$$\kappa \geq \frac{50 + 22n}{n - 70}.$$

Luego, se puede elegir $\kappa = 1612 = 50 + 22 \cdot 71$, y $K = 71$, con lo cual se demuestra que el algoritmo pertenece a $O(n)$.