

Cálculo de eficiencia para algoritmo para mezclar k vectores de tamaño n , usando divide y vencerás y recursividad.

Teóricamente, vamos a suponer que k es una potencia de 2. $k = 2^m$, por lo que mezclaríamos las $k/2$ parejas de vectores (1-2,3-4,5-6...) de longitud n . Una vez mezcladas, se mezclarían las $k/4$ parejas de vectores, de longitud $2n$: (1-2 con 3-4, 5-6 con 7-8...), y así sucesivamente.

Para calcular la eficiencia vamos a usar expansión.

Tenemos que $T(k) = 2T(k/2) + c_2nk$.

$2T(k/2)$ por que llamamos recursivamente a la función dos veces, pasándole $k/2$ como argumento.

c_2nk por que la acción de combinar dos vectores es lineal.

Vamos a calcular $T(k/2)$.

$T(k/2) = 2T(k/4) + c_2n(k/2)$.

Entonces, $T(k) = 4T(k/4) + c_22nk$.

También $T(k) = 8T(k/8) + c_23nk$.

Se puede concluir que $T(k) = 2^i T(k/2^i) + c_2ink$.

Al llegar al final de la recurrencia, va a haber un momento en el cual en el lado derecho tengamos $T(1)$.

Ésto quiere decir, que cuando $i = m$, tendremos que $T(k) = 2^m T(1) + c_2mnk$.

Sabiendo que $2^m = k$, y $m = \log_2 k$, y tomando $T(1)$ como c_1 , tenemos lo siguiente:

$T(k) = c_1k + c_2nk \log_2 k$.

$nk \log_2 k$ hace despreciable k , por lo que tenemos que la eficiencia es $O(\log(k) * nk)$.