Soma dos elementos do vetor

1. divide o vetor em duas partes $n/2$
2. conquista recursivamente cada metade
3. combina as duas soluções encontradas

$$T(n) = 2T(n/2) + f(n)$$

$f(n)$ custo da operação de soma e cálculo do índice do meio

Teorema Mestre:

$a = 2$

$b = 2$

$f(n) = \Theta(1) = c$

$n^{\log_b a} = n^{\log_2 2} = n^1$

caso 1 TM:

$f(n) \in O(n^{\log_b a - \varepsilon})$ para algum $\varepsilon > 0$?

seja $\varepsilon = 1$

$n^{\log_b a - \varepsilon} = n^{1-1} = n^0 = 1$

$f(n) = O(1)$

$n^0 = 1 = \Theta(1)$

$$\Theta(1) \in \Theta(1)$$

se $f(n) \in O(n^{\log_b a - \varepsilon})$ então $T(n) = \Theta(n^{\log_b a})$

$$T(n) = \Theta(n)$$

mesma complexidade da forma trivial

valor de um elemento máximo em vetor

relação de recorrência:

- $n = r - p + 1$ o tamanho do vetor
- $C(n)$ comparações

$n = 1 \to C(1) = 0$

$n > 1 \to$ chama rec. à esquerda $n/2$
$\to$ chama rec. à direita $n/2$
$\to$ compara $1x$

$$C(n) = C(n/2) + C(n/2) + 1$$
$$= 2C(n/2) + 1$$

teorema mestre

$a = 2$
$b = 2$
$f(n) = 1 = \Theta(1)$

idem caso anterior

Kendall tau

recorrência

$T(n) = \text{Dividir}(n) + \text{Conquista}(n) + \text{combinar}(n)$

$T(n) = a\,T(n/b) + f(n)$

L subproblemas

L tamanho

L etapas divisão e combinação

merge sort:

$T(1) = \Theta(1)$

$T(n) = 2T(n/2) + \Theta(n)$

a = 2 chamadas recursivas

b = cada chamada tamanho $n/2$

$f(n) = \Theta(n)$ percorre os elementos uma vez

Teorema mestre

$n^{\log_b a} = n^{\log_2 2} = n^1$

caso 1:

$f(n) = \Theta(n) \in O(n^{1-\varepsilon})$, para $\varepsilon > 0$?

não! pois $1-\varepsilon,\ \varepsilon > 0 < 1$

caso 2:

$\Theta(n) \in \Theta(n^1)$ sim!

então
$T(n) \in \Theta(n \log n)$
comportamento do merge sort

closest points

merge sort: $O(n \log n)$

em closest_pair_rec, em n pontos

1- divisão:
$2 \cdot T(n/2)$

2. separar py em Q/R $O(n)$

construção faixa: $O(n)$

percorrer faixa: $n \cdot 7 = O(n)$

$f(n) = O(n)$

mesmo caso acima

$T(n) = \Theta(n \log n)$