# QuickSort

## Partição de Vetor

Resolver o problema da Partição de Vetor serve para projetar o Quicksort.

#### Partição de Vetor

Entrada: (v, a, b) onde v é um vetor indexado por [a..b].

Saída : Seja x = v[b]. Modifica o vetor v de forma a garantir a

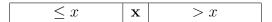
existência de um índice  $m \in [a-1..b]$  tal que

$$v[i] \le x, \forall i \in [a..m],$$

е

$$x < v[i], \forall i \in [m+1..b].$$

Devolve este índice m.



• x é chamado de  $piv\hat{o}$ .

#### Particiona(v, a, b)

$$\begin{aligned} m &\leftarrow a - 1 \\ i &\leftarrow a \\ \text{Para } i \leftarrow a \text{ to } b \\ \text{Se } v[i] &\leq x \\ m &\leftarrow m + 1 \\ \text{Troca}(v, m, i) \end{aligned}$$

 ${\sf De}_{\sf Volva}\ m$ 

Executar Particiona(v, 1, 6)

Importante: após a execução do Particiona, x fica na posição que deveria estar se o vetor estivesse ordenado.

**Teorema 1.** Particiona executa em tempo  $\Theta(n)$ , onde n é o tamanho do vetor.

### Quicksort

É um algoritmo projetado usando a técnica "divisão e conquista".

 $\begin{aligned} & \text{Quicksort}(v, a, b) \\ & \text{Se } a < b \\ & m \leftarrow \text{Particiona}(v, a, b) \\ & \text{Quicksort}(v, a, m - 1) \\ & \text{Quicksort}(v, m + 1, b) \end{aligned}$ 

#### Análise

T(n): tempo do Quicksort em vetores de n elementos.

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1 \\ T(k) + T(n-k-1) + \Theta(n), & Se \ n > 1 \end{cases}$$

onde  $\Theta(n)$  vem do tempo do Particiona.

#### Um Caso Ruim

Ocorre quando a Particiona produz as duas partições com tamanho 0 e n-1. Isso ocorre, **por exemplo**, quando vetor JÁ está ordenado. Mas existem outros casos ruins também.

Assim, a relação de recorrência fica:

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1\\ T(0) + T(n-1) + \Theta(n), & Se \ n > 1 \end{cases}$$

ou

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1 \\ T(n-1) + \Theta(n), & Se \ n > 1 \end{cases}$$

É possível mostrar que a solução dessa recorrência é  $T(n) = \Theta(n^2)$ .

#### Pior Caso

Como saber que o caso ruim é o pior caso?

Teorema 2. A recorrência

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1\\ \max_{0 \le k \le n-1} \{ T(k) + T(n-k-1) \} + \Theta(n), & Se \ n > 1 \end{cases}$$

tem como solução  $T(n) = O(n^2)$ .

Demonstração. (indução em n)

base: (Exercício)

**hipótese:**  $T(q) \le cq^2, \forall 1 \le q < n$ 

passo: Vamos provar que  $T(n) \le cn^2$ .

$$T(n) = \max_{0 \le k \le n-1} \left\{ T(k) + T(n-k-1) \right\} + bn$$

$$\le \max_{0 \le k \le n-1} \left\{ ck^2 + c(n-k-1)^2 \right\} + bn$$

$$= c \max_{0 \le k \le n-1} \left\{ k^2 + (n-k-1)^2 \right\} + bn$$

A expressão  $k^2+(n-k-1)^2$  atinge valor máximo quando k=0 ou k=n-1 [Exercício CLRS]. Assim,

$$T(n) \le c \max_{0 \le k \le n-1} \left\{ k^2 + (n-k-1)^2 \right\} + bn$$
  
=  $c(n-1)^2 + bn$   
 $\le cn^2$ ,

onde a última desigualdade é válida se  $c \geq b$  e  $n \geq 1$ .

A prova mostra que quando Particiona produz as partições com tamanho 0 e n-1 é de fato o **pior caso**.

#### Algumas Conclusões Até Agora...

A complexidade de tempo do Quicksort no **pior caso** é  $\Theta(n^2)$ .

A complexidade de tempo do Quicksort é  $O(n^2)$ .

#### Melhor caso

Não seremos rigorosos na análise de melhor caso (só daremos a ideia). Ocorre quando a Particiona produz as duas partições com tamanho "igual".

$$\lfloor n/2 \rfloor$$
  $\mathbf{x}$   $\lceil n/2 \rceil - 1$ 

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1\\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil - 1) + \Theta(n), & Se \ n > 1 \end{cases}$$

Teorema 3. A relação de recorrência:

$$T(n) = \begin{cases} \Theta(1), & Se \ n \le 1\\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil - 1) + \Theta(n), & Se \ n > 1 \end{cases}$$

tem como solução  $T(n) = \Theta(n \log n)$ .

Demonstração. (Exercício)

#### Outra Conclusões...

A complexidade de tempo do Quicksort no **melhor caso** é  $\Theta(n \log n)$ .

A complexidade de tempo do Quicksort é  $\Omega(n \log n)$ .

Na prática: O Quicksort é um dos mais eficientes algoritmos de ordenação.

Na teoria: Seu pior caso é aproximadamente igual aos dos algoritmos mais simples de ordenação.