

Análise e Projeto de Algoritmos

Divisão e Conquista (Parte 2)

Prof. Bruno Bruck



Quick Sort

- Problema

- ▣ Ordenar um vetor

- Ideia Básica

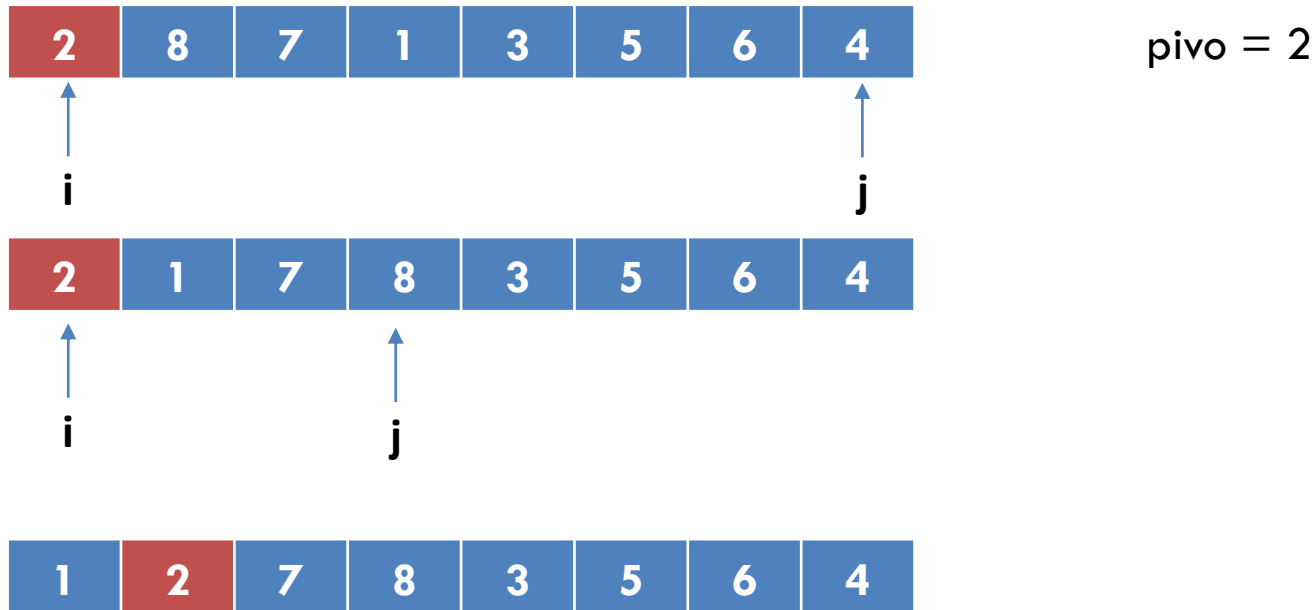
- ▣ Usar a técnica de divisão e conquista
 - ▣ Selecionar um pivô, dividir o vetor em 2 partições
 - Uma contendo os elementos com valor maior do que o pivô
 - E a outra contendo os elementos com valor menor ou igual que o pivô

Quick Sort

- Implementação baseada em 2 funções
 - ▣ quickSort
 - ▣ partition

Quick Sort

- Exemplo de funcionamento (Partition):
 - ▣ Somente uma chamada da função partition



Quick Sort

□ Pseudocódigo

```
partition(A, inicio, fim)
1.  pivo ← A[inicio]
2.  i ← inicio - 1
3.  j ← fim + 1
3.  while true
4.      do
5.          i ← i + 1
6.          while A[i] ≤ pivo
7.
8.      do
9.          j ← j - 1
10.     while A[j] > pivo
11.
12.     if i ≥ j
14.         break;
15.     swap(A[i], A[j])
16. swap(A[inicio], A[j])
```

Quick Sort

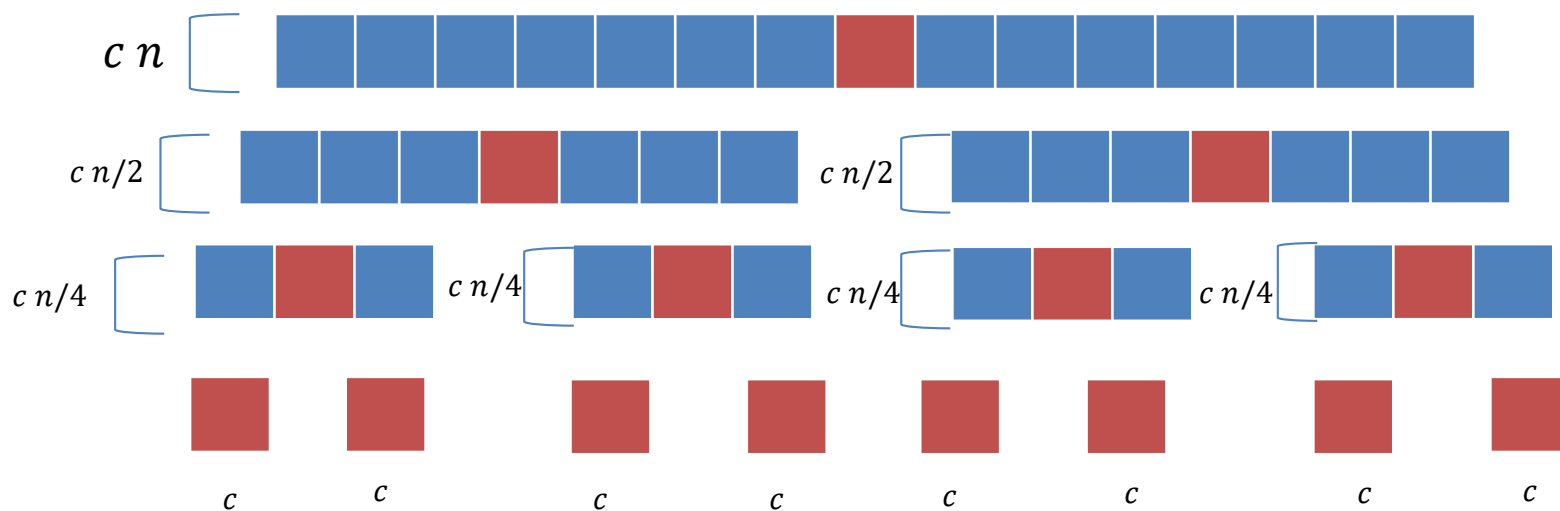
□ Pseudocódigo

quickSort(A, l, r)

1. **if** $l < r$ **then**
2. $q \leftarrow$ **Partition** (A, l, r)
3. **quickSort** (A, l, q-1)
4. **quickSort** (A, q+1, r)

Quick Sort

□ Árvore de Recursão (Melhor caso)



Nível 0: cn

Nível 1: $2 \left(c \frac{n}{2} \right) \Rightarrow cn$

Nível 2: $4 \left(c \frac{n}{4} \right) \Rightarrow cn$

Nível N: $n \left(c \frac{n}{n} \right) \Rightarrow cn$

$$\begin{cases} T(1) = 1 \\ T(n) = T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + cn \end{cases}$$

Quick Sort

□ Árvore de Recursão (Pior Caso)

$$\begin{cases} T(1) = 1 \\ T(n) = T(n - 1) + c n \end{cases}$$

