

## **Ordenação de dados: Divisão e conquista**

# Retomando...

- Classificação e ordenação de dados
  - BubbleSort
  - Ordenação por seleção
  - Ordenação por inserção

## Quick Sort



# Quick Sort

- Um dos algoritmos de ordenação interna mais rápidos que se conhece para uma ampla variedade de situações;
  - Provavelmente é um dos mais utilizados.
- A ideia básica é dividir um conjunto com  $N$  itens em subconjuntos menores;
- Os subconjuntos são ordenados de forma independente dos demais.



# Quick Sort

- A parte mais delicada do método é o processo de partição;
- O vetor  $A[n]$  é rearranjado por meio da escolha arbitrária de um **pivô**  $x$ ;
- O vetor  $A$  é dividido em duas partes (ou mais):
  - Parte esquerda: índices  $\leq x$ .
  - Parte direita: índices  $\geq x$ .

# Quick Sort - partição

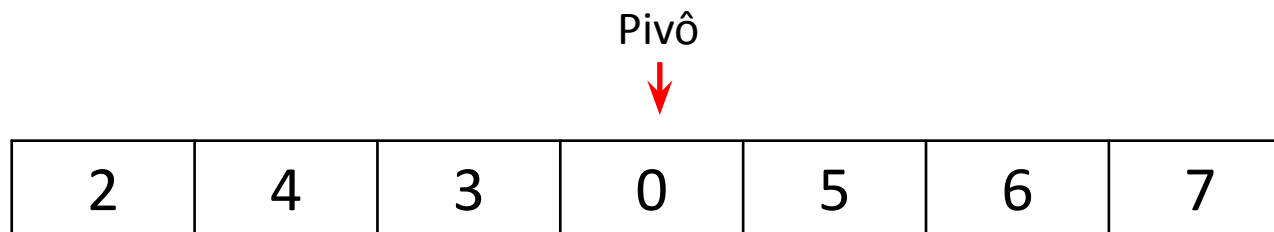
- Algoritmo de ordenação:
  - 1. Escolha arbitrariamente um **pivô** x;
  - 2. Usando uma variável i, percorra o vetor a partir da esquerda até que  $A[i] \geq A[x]$ ;
  - 3. Usando uma variável j, percorra o vetor a partir da direita até que  $A[j] \leq A[x]$ ;
  - 4. Troque  $A[i]$  com  $A[j]$ ;
  - 5. Continue este processo até os apontadores i e j se cruzarem;
  - 6. Repita estes passos para o subconjunto da esquerda e da direita.

i			Pivô			j
↓			↓			↓
0	1	2	3	4	5	6
7	4	6	5	0	3	2



# Quick Sort – depois da partição

- Ao final o vetor A está organizado de tal forma que:
  - Os itens em  $A[\text{Esq}]$ ,  $A[\text{Esq} + 1]$ , ...,  $A[j]$  são menores ou iguais a  $x$ ;
  - Os itens em  $A[i]$ ,  $A[i + 1]$ , ...,  $A[\text{Dir}]$  são maiores ou iguais a  $x$ .
- **ATENÇÃO:** os elementos dentro de seus subconjuntos (Esq e Dir) ainda não estão ordenados entre si.





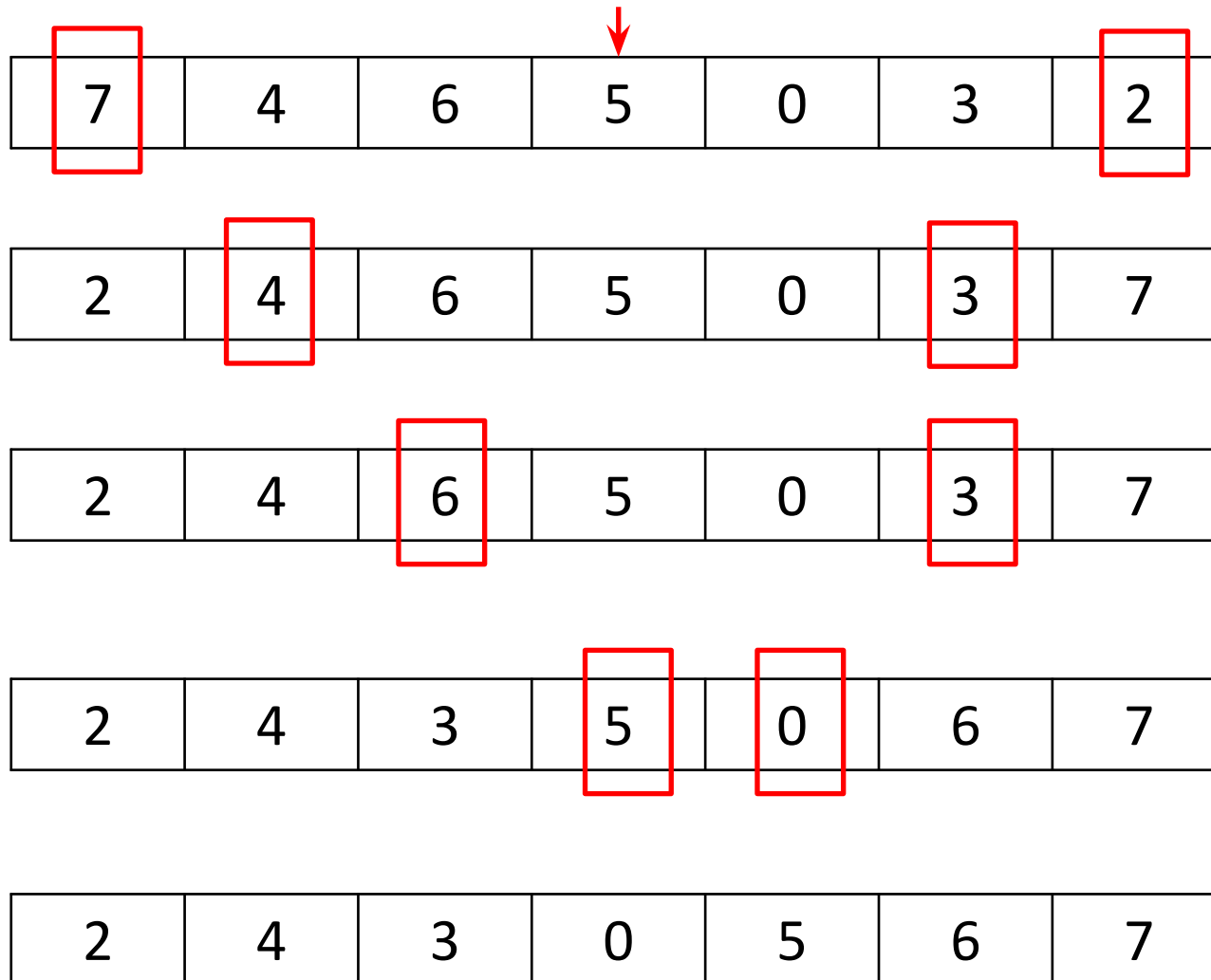
# Quick Sort - Exemplo

- O pivô  $x$  é escolhido:
  - Elemento central:  $A[(i + j) / 2]$ .
- Exemplo:

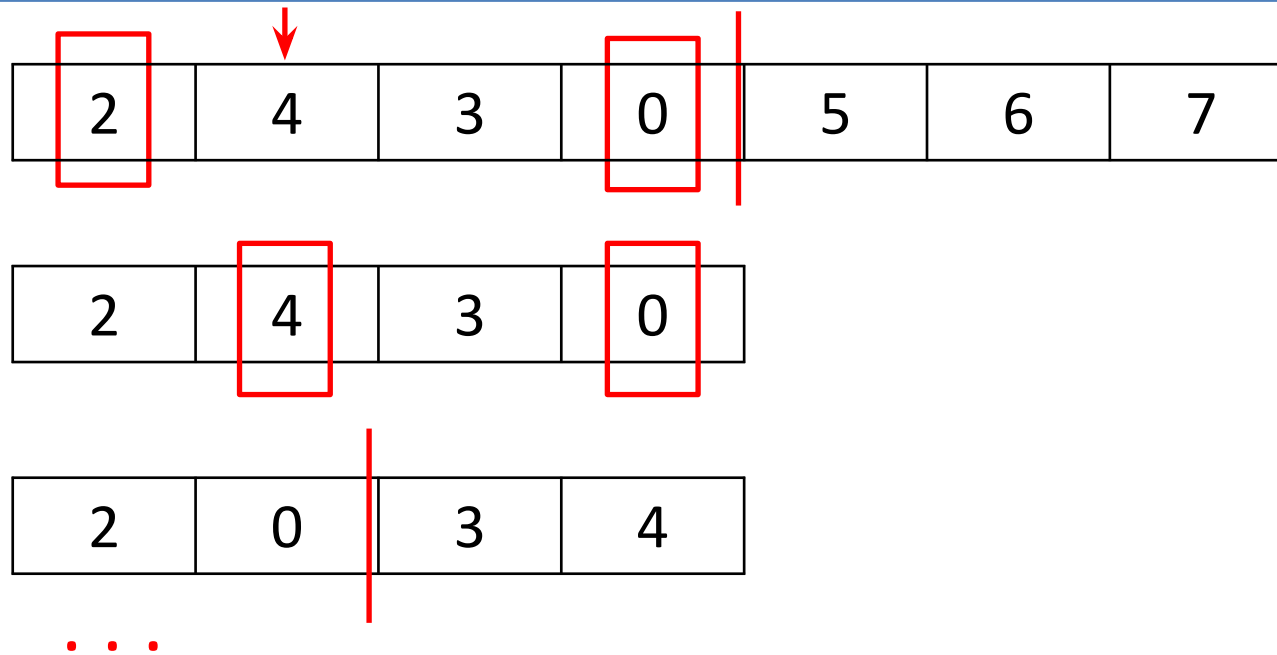
0	1	2	3	4	5	6
7	4	6	5	0	3	2



# Quick Sort – Primeira etapa



## Quick Sort – Segunda etapa



- Este processo é repetido recursivamente até que toda a lista esteja em ordem.

# Ordenação de dados

## ***Merge Sort***



# Merge Sort

- Segue a mesma ideia do *Quick sort*:
  - Ordenação por segmentação: divisão dos elementos em subconjuntos para ordenação;
- A ideia básica é dividir um conjunto com  $N$  itens em subconjuntos menores;
- Os subconjuntos são ordenados de forma independente dos demais.



# Merge Sort

- Assumindo que um conjunto com apenas um elemento já está ordenado:
  - Dividir a lista ao meio;
  - Ordenar o lado esquerdo da lista (ignorar o lado direito);
  - Ordenar o lado direito da lista (ignorar o lado esquerdo);
  - Mesclar (*merge*) as duas metades.



# Merge Sort

1. Encontrar um elemento pivô central e dividir o conjunto em subconjuntos;
2. Repetir esta etapa até que se tenha subconjuntos de um elemento;
3. Ordenar subconjuntos adjacentes;
4. Juntar subconjuntos ordenados com os demais.



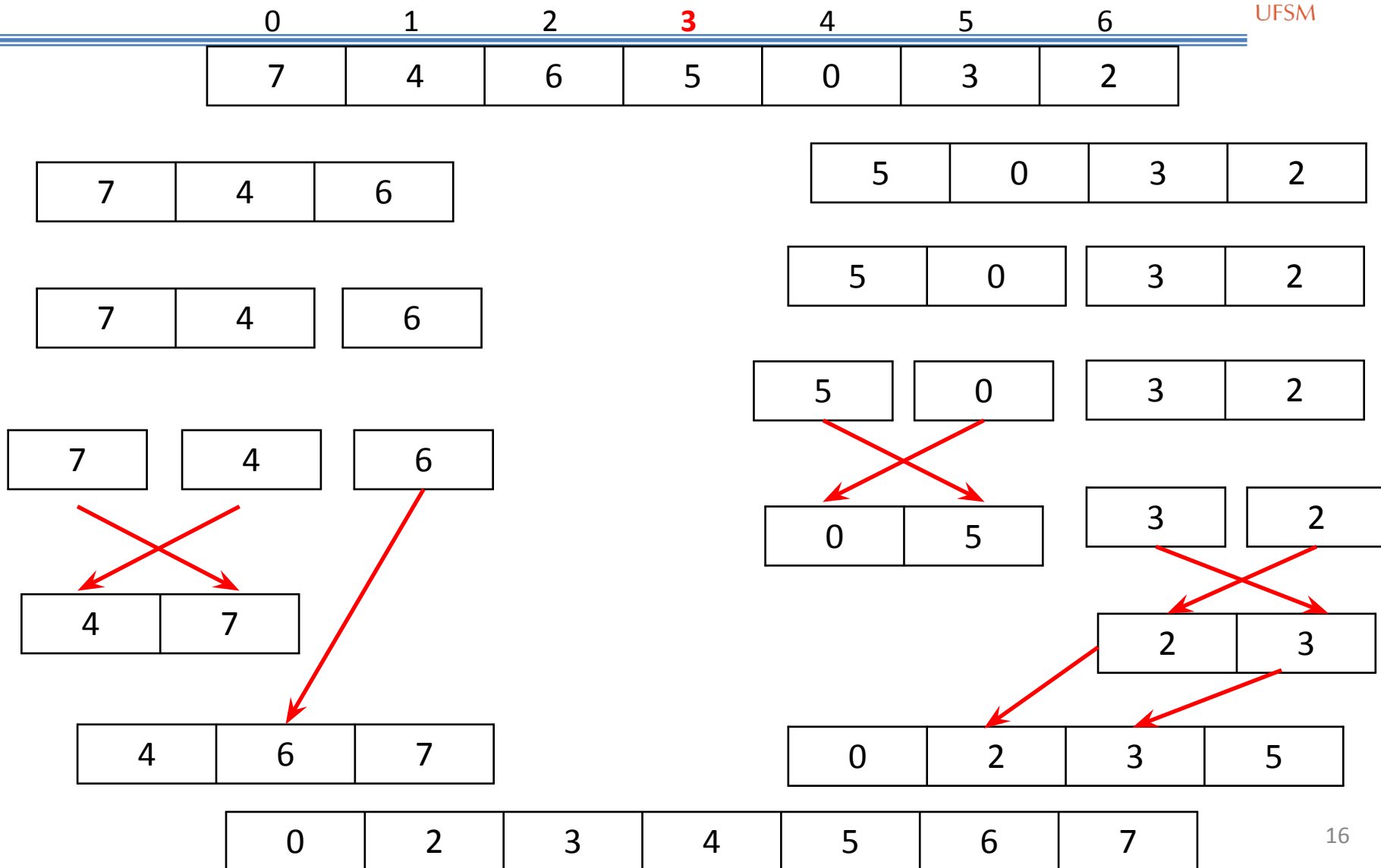
# Merge Sort

- O pivô  $x$  é escolhido:
  - Elemento central:  $A[(i + j) / 2]$ .
- Exemplo:

0	1	2	3	4	5	6
7	4	6	5	0	3	2



# Merge Sort







# Exercício

- Desenvolva um algoritmo que coloque em ordem crescente os elementos de um vetor, através da classificação Quick Sort;
- Desenvolva um algoritmo que coloque em ordem crescente os elementos de um conjunto, através da classificação Merge Sort.

[15 sorting algorithms animation](#)