

**ESTRUTURA DE DADOS**  
**ALGORITMO DE ORDENAÇÃO RÁPIDA (QUICKSORT)**

**Prof. Enyo José**

# Quicksort

- Complexidade:
  - Melhor caso e caso médio é  $n \log n$  ( $n$  do particionamento e  $\log n$  da divisão)
  - Pior caso é  $n^2$
- Lógica:
  - Quebra/divide o vetor em duas partes sucessivamente (Particiona)
  - Em seguida faz uma ordenação parcial
  - Usa recursividade para fazer as análises das duas partes do vetor

# Quicksort - Funcionamento

- Escolha um pivô (elemento a ser ordenado)
  - Os elementos maiores que ele ficarão a sua direita e os menores que ele ficarão a sua esquerda
  - Não importa a ordem dos sub-vetores esquerda e direita, a única preocupação é que os elementos estejam do lado certo do pivô

0	1	2	3	4	5
3	0	1	5	8	6

**pivô**

- A escolha do pivô pode ser feita em qualquer posição do vetor
  - O vetor está desordenado inicialmente, então o valor de cada posição pode ser qualquer um
  - Vamos utilizar o valor da posição 0 como pivô

# Quicksort - Funcionamento

- Depois da escolha do pivô, o vetor é percorrido por duas variáveis
  - Uma delas **começa da esquerda (esq)** e vai sendo **incrementada para a direita**
  - A outra **começa da direita (dir)** e vai sendo **decrementada para a esquerda**
- O objetivo é encontrar elementos a serem trocados, ou seja:
  - Maiores que o pivô na esquerda
  - Menores que o pivô na direita

# Quicksort - Funcionamento

- A condição de parada é que *esq* e *dir* se encontrem (cruzem).
  - O local que eles pararem no vetor indica a posição que o valor do pivô deve ser colocado
  - Note que nem sempre é o meio do vetor (posição central), pois início e fim do vetor vai mudando a medida que as chamadas recursivas são feitas
- Por fim, troca-se o valor de *dir* com o do pivô
  - Neste momento o pivô estará no local correto dele: a esquerda valores menores que ele e a direita valores maiores
- Os elementos da esquerda e da direita não estarão obrigatoriamente ordenados entre si
  - Por conta disto, são feitas chamadas recursivas para as duas porções do vetor (esquerda e direita do pivô)

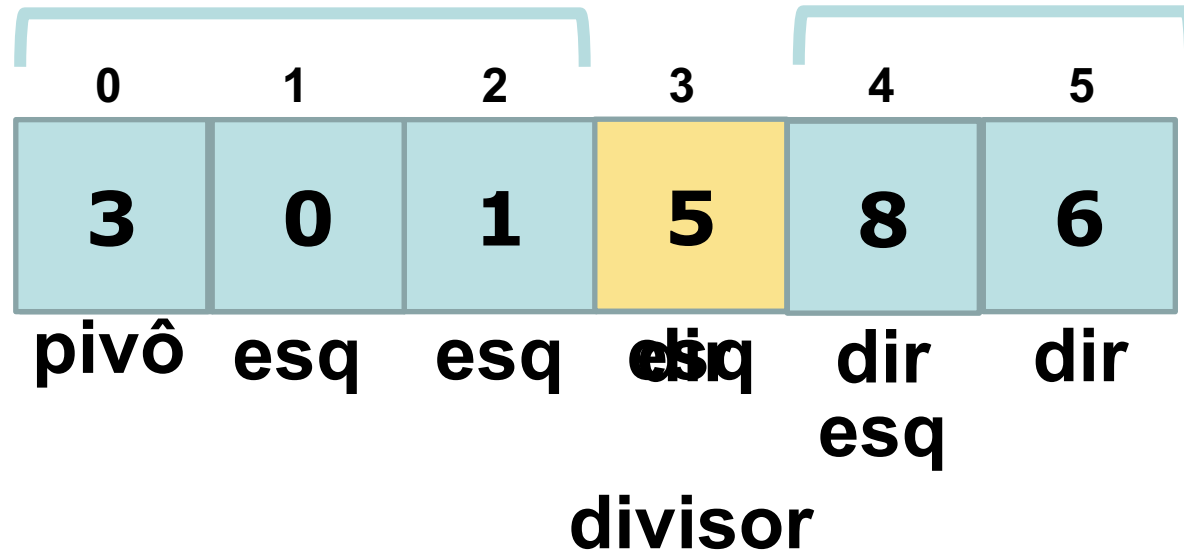
# Quicksort - Funcionamento

- O procedimento particiona é chamado seguido de novas chamadas recursivas e assim por diante...
  - A condição de parada da recursividade é quando início = fim

# Ilustração

- A ilustração dos próximos slides está disponível em video em: [https://youtu.be/kfRL4\\_-MxJU](https://youtu.be/kfRL4_-MxJU)

## Chamada recursiva 1 Chamada recursiva 2



Temp 8      Temp 5