

# Ordenação: Quicksort

Estrutura de Dados e Algoritmos – Ciência da Computação



**INSTITUTO  
FEDERAL**  
Brasília

Prof. Daniel Saad Nogueira  
Nunes

IFB – Instituto Federal de Brasília,  
Campus Taguatinga



# Sumário

---

1 Quicksort

2 Análise



# Sumário

---

## 1 Quicksort



# Quicksort

---

## Quicksort

O Quicksort se baseia na escolha de um pivô. Após escolhido este pivô, a sequência original é particionada em três partes:

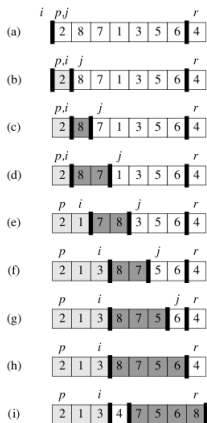
- 1 Elementos menores ou iguais que o pivô;
- 2 Pivô;
- 3 Elementos maiores que o pivô;

O procedimento é aplicado recursivamente na primeira e última partes.



# Quicksort

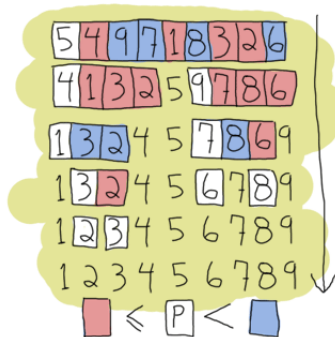
## Exemplo





# Quicksort

## Exemplo





# Mergesort

---

```
23 void quick_sort(int *v, size_t size){  
24     quick_sort_helper(v,0,size-1);  
25 }
```



# Mergesort

---

```
15 void quick_sort_helper(int*v, int l,int r){
16     if (l < r){
17         size_t pos = partition(v,l,r,r);
18         quick_sort_helper(v, l, pos - 1);
19         quick_sort_helper(v, pos + 1, r);
20     }
21 }
```





# Mergesort

```
1 static size_t partition(int *vet, int esq, int dir, int pivot){
2     size_t pos, i;
3     swap(vet, pivot, dir);
4     pos = esq;
5     for(i = esq; i < dir; i++){
6         if (vet[i] < vet[dir]){
7             swap(vet, i, pos);
8             pos++;
9         }
10    }
11    swap(vet, dir, pos);
12    return pos;
13 }
```



# Sumário

---

## 2 Análise



## Quicksort: Análise

---

### Análise

A relação de recorrência do Mergesort, no pior caso, corresponde à:

$$T(n) = T(n - 1) + O(n) \in \Theta(n^2)$$

Contanto, no caso médio, o Quicksort divide as partições de modo em que a primeira e a última partição tenham tamanhos similares, o que leva a uma relação de recorrência que se aproxima de:

$$T(n) = 2 \cdot T(n/2) + O(n) \in \Theta(n \lg n)$$

In-place	Estável
X	X



# Quicksort

## Análise

