

## Sorting Algorithms



Retirado de (<https://embed-ssl.wistia.com/deliveries/70d6f4e10e2badb5ef394f00c17ad2bc1c14f6e7.jpg>), em 18/08/2021

## Atividade Busca Binária

## Quick Sort e Busca Binária Recursiva

- Implemente a função do *Quick Sort* para ordenar as informações de um grupo de pessoas. **A ordenação será alfabética crescente.**
- Crie uma estrutura de dados chamada **PESSOA**, que contém o **nome** e a **idade** de uma pessoa.
- Crie um vetor para armazenar os dados de **N** pessoas que serão ordenados.
- O valor da pessoa que deve ser buscada, assim como os demais elementos que estão presentes no vetor devem ser *hardcodados* no código.

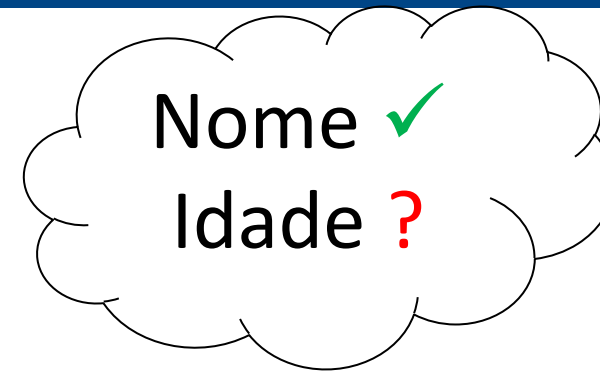
# Quick Sort e Busca Binária Recursiva

nome	idade
Gohan	23
Goku	43
Chichi	43
Goten	13
Vegeta	48
Bulma	47
Trunks	14
Mr. Satan	44
Sanji	19
Chopper	19
Robin	28
Franky	34
Brook	88

nome	idade
All Might	49
Tomura	20
Aizawa	31
Hawks	23
Midoriya	16
Jeanist	36
Togata	18
Endeavour	46
Dabi	24
Uraraka	15
Kirishima	15
Todoroki	15
Bakugo	15

# Busca Binária Recursiva

```
typedef struct{  
    char nome[80];  
    int idade;  
} Pessoa;
```



```
int BuscaBinariaRecursiva(  
    Pessoa listaDePessoas[], Pessoa *p,  
        int L, int H);
```

Na main:

```
Pessoa perdido, listaDePessoas[100];  
.  
.  
.  
if (BuscaBinariaRecursiva(listaDePessoas,  
                            &perdido, 0, n-1) != -1)  
    Achou  
else  
    Não achou
```