

PO (Pesquisa e Ordenação)

↳ Java

↳ POO (programação orientado objeto)

↳ Pesquisa: Métodos de busca e árvores
 não binária (string e números)

Algoritmos: Busca Binária, Trie, N-óreo,

B+, B

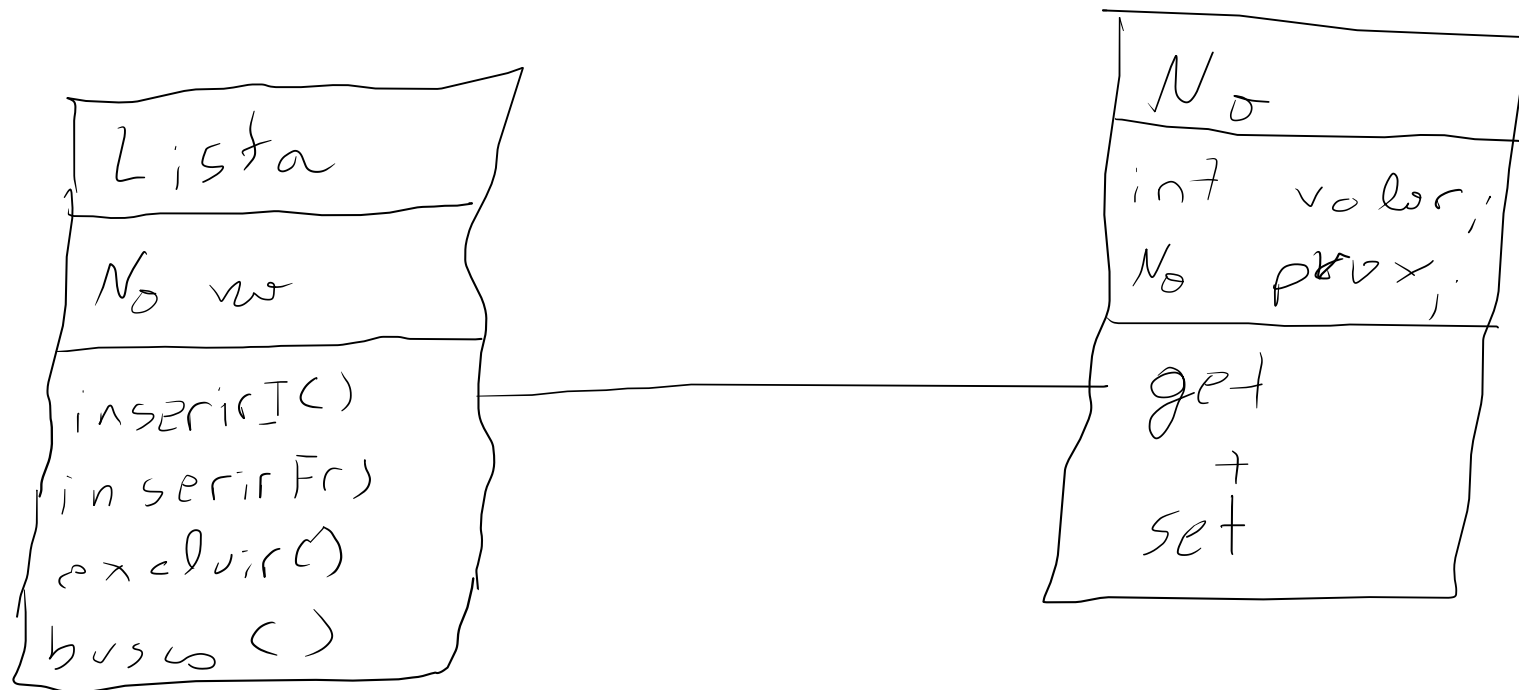
↳ Ordenação: Bubble, Merge Sort, Radix

Lista usando classes

```
struct Lista {  
    int valor;  
    Lista *prox;  
};
```

}
C++
classe

Java:

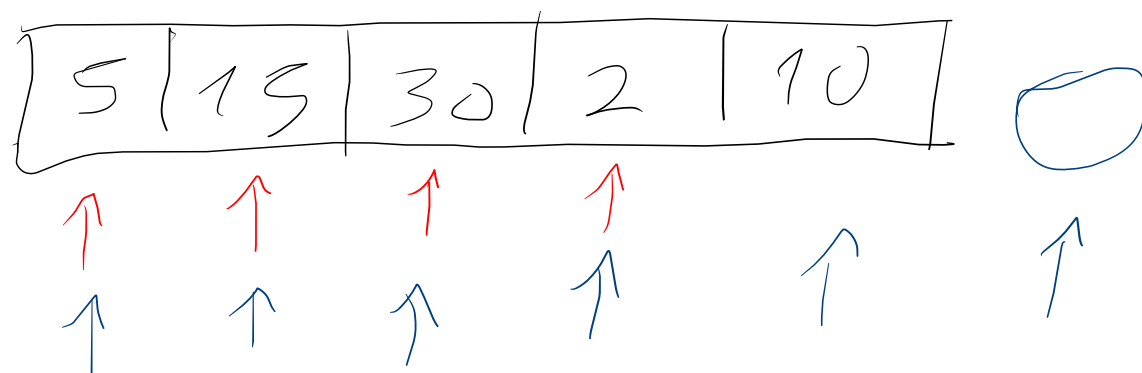


Tarefa:

- Lista ordenada
- Filas
- Pilhas

Buscos

1. Exaustivo / Linear



$$19 = 5$$

$$19 = 15$$

$$19 = 30$$

$$19 = 2$$

$$19 = 10$$

Não o chav

$$2 = 5$$

$$2 = 15$$

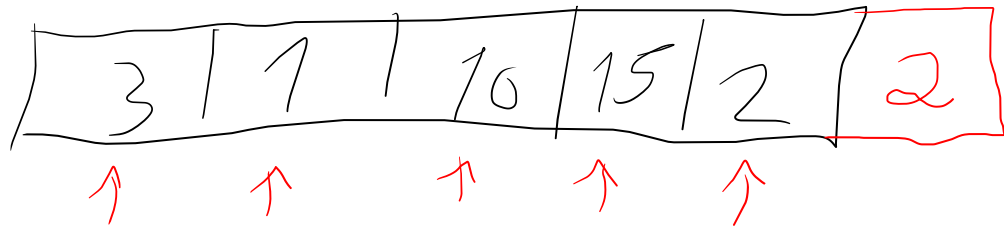
$$2 = 30$$

$$2 = 2 \quad \checkmark$$

Busca Linear:

- Passo por cada posição do array (comparando)
- Se chegou no final do array: Não achou
- Senão: Achou

2. Sentinela



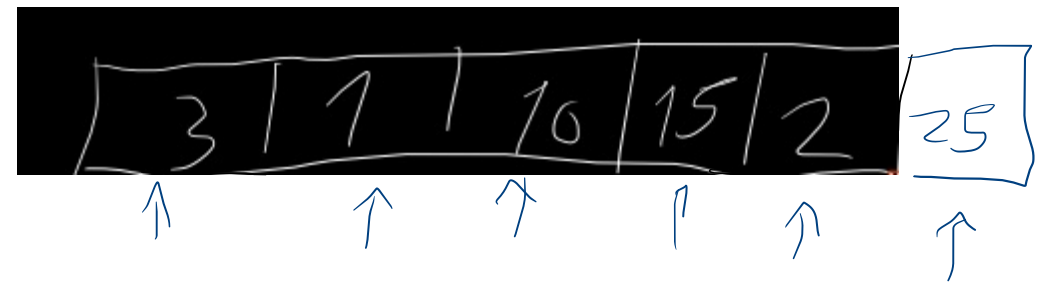
$$2 \neq 3$$

$$2 \neq 1$$

$$2 \neq 10$$

$$2 \neq 15$$

$$2 = 2 \quad \checkmark$$



$$25 \neq 3$$

$$25 \neq 1$$

$$25 \neq 10$$

$$25 \neq 15$$

$$25 \neq 2$$

$$25 = 25 \quad (\text{Falso, o che sentinel})$$

Busca Sentinela:

- Adicionar o sentinela (valor no final)
- Passar pelas posições do array (comprimento)
- Se achou o sentinela; Não existe
- Senão; Achou

3. Indexada

Observação: Só vale para vetores ordenados

Vantagem: Se em algum momento o meu valor for menor do que a posição eu posso parar a busca.

$$\text{arr}[0] \leq \text{arr}[1] \leq \text{arr}[2] \dots \leq \text{arr}[n]$$
$$20 \leq \text{arr}[2]$$

1	3	25	30	31	100
---	---	----	----	----	-----

↑ ↑ ↑ ↑ ↑

$$31 > 1$$

$$31 > 3$$

$$31 > 25$$

$$31 > 30$$

$$31 > 31 \text{ (Paro)} \Rightarrow 31 = 31 \quad \checkmark$$

1	3	25	30	31	100
---	---	----	----	----	-----

↑ ↑ ↑

$$17 > 1$$

$$17 > 3$$

$$17 > 25 \text{ (Paro)} \Rightarrow 17 = 25 \text{ (Falso)}$$

$$\uparrow$$

101

Busca Indexada

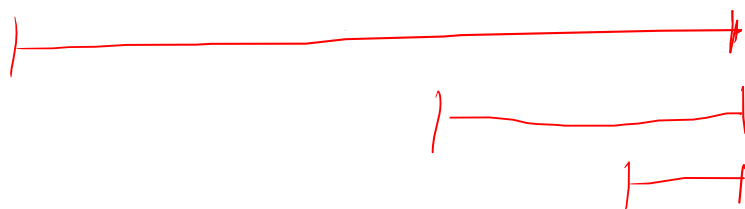
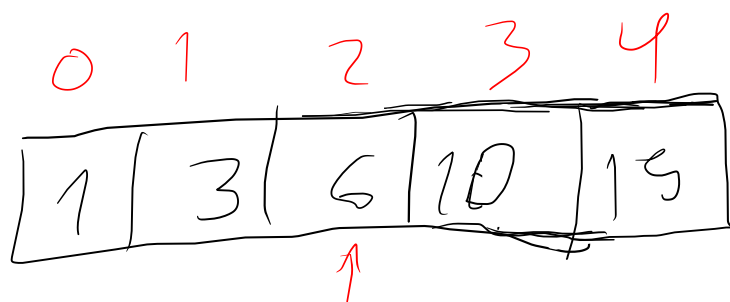
- Passo por todas as posições que $arr[i]$ é menor que o buscado
- se parar e for igual: achou
- Se não: não achou

4. Busca Binária

Observação: O array tem que estar ordenado.

Vantagem: Busca mais rápida $O(\log n)$

$ini \leq fim$



$$15 = 6$$

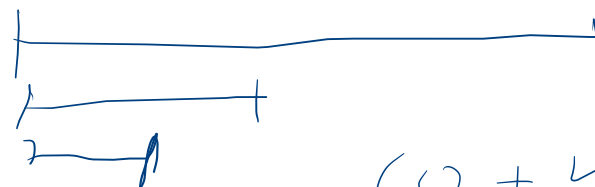
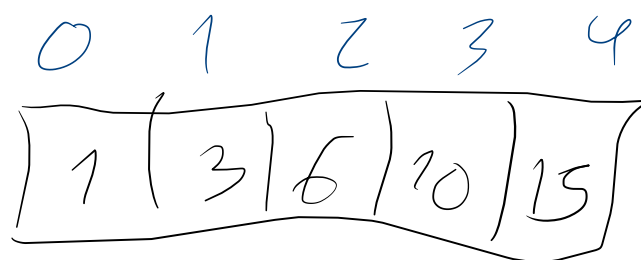
$$(0 + 4) / 2 = 2$$

$$15 = 10$$

$$(3 + 4) / 2 = 3$$

$$15 = 15 \checkmark$$

$$(4 + 4) / 2 = 4$$



$$0 = 6$$

$$0 = 1$$

$$0 = 1$$

$$(0 + 4) / 2 = 2$$

$$(0 + 1) / 2 = 0$$

$$(0 + 0) / 2 = 0$$

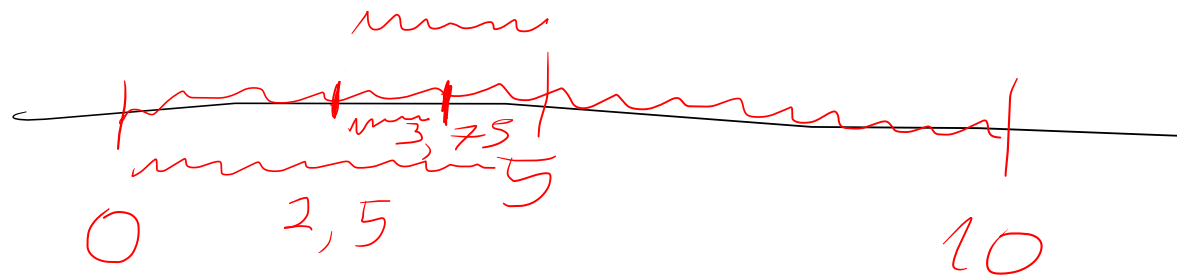
$$(0 + -1)$$

Busca Binária:

- Definir início e fim
- Verificar o meio do intervalo:
 - Se for igual: ache
 - Se for maior: mude início
 - Se for menor: mude o meio
- Se $\text{início} > \text{fim}$, para: Não ache

Aplicação: Achar raíz quadrada,

$$\sqrt{9} = ? \quad y \Rightarrow y^2 = 9$$



$$5^2 > 9 \quad (V)$$

$$2,5^2 > 9 \quad (F)$$

$$3,75^2 > 9 \quad (V)$$

$$3,125^2 > 9 \quad (V)$$

$$2,81^2 > 9 \quad (F)$$

$$x > y \Rightarrow \sqrt{x} > \sqrt{y}$$

$$100 > 25 \Rightarrow \sqrt{100} = 10 > \sqrt{25} = 5$$

$$[0, 10], [0, 5], [2.5, 5], [2.5, 3.75], [2.5, 3.125],$$

$$[2.81, 3.125]$$

Tarefa: Implementar os passos em lista.