Fundamentos da Programação

Pesquisa e Ordenação

P.PORTO
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Conteúdo

- Algoritmos de Pesquisa
- Algoritmos de Ordenação

Pesquisa

Algoritmos de pesquisa

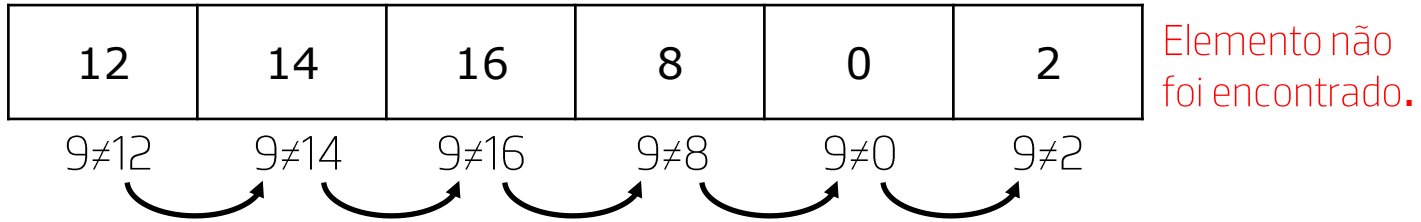
- Utilizam-se para **localizar um elemento** numa coleção de dados.

Pesquisa Sequencial

- É uma técnica simples para pesquisa de elementos num vetor.
- Através de um ciclo, percorre-se sequencialmente cada posição do vetor começando na primeira posição.
- Compara-se o elemento em cada posição com o elemento que se procura.
- A pesquisa termina quando:
  - Se chega ao fim do vetor sem encontrar o elemento.
  - Se encontra o elemento.

Pesquisa Sequencial

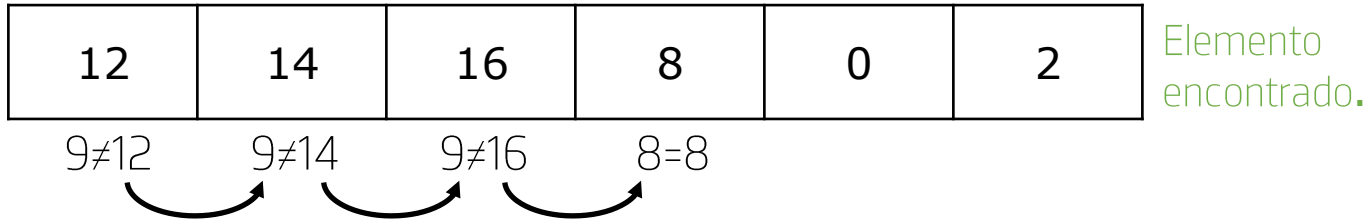
- Pesquisa do valor **9**:



Elemento não foi encontrado.

Pesquisa Sequencial

- Pesquisa do valor **8**:

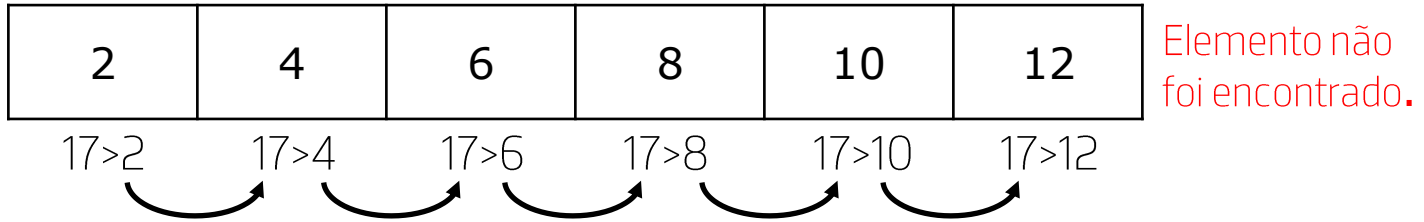


Pesquisa Sequencial Ordenada

- É uma evolução da pesquisa sequencial. Assume que os elementos do vetor estão ordenados de forma crescente.
- Através de um ciclo, percorre-se sequencialmente cada posição do vetor começando na primeira posição.
- A pesquisa termina quando:
  - Se chega ao fim do vetor.
  - Se encontra no vetor um elemento maior que o elemento procurado.
  - Se encontra o elemento.

Pesquisa Sequencial Ordenada

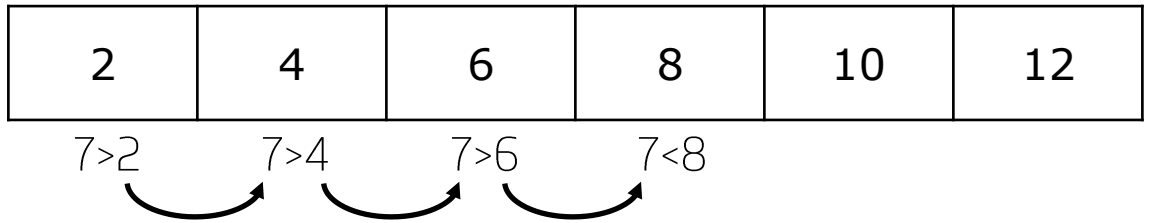
- Pesquisa do valor **17**:



Elemento não foi encontrado.

Pesquisa Sequencial Ordenada

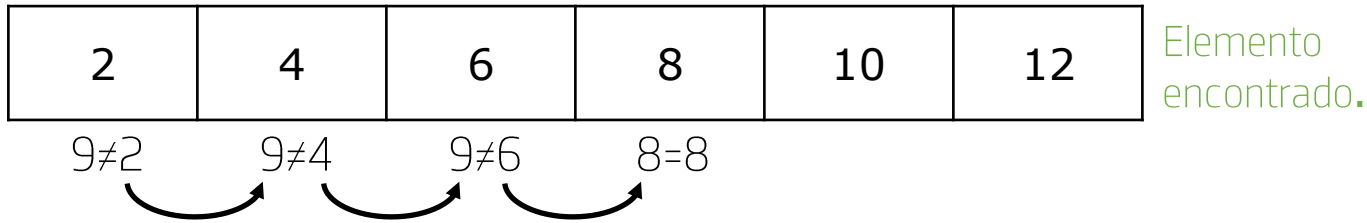
- Pesquisa do valor **7**:



Elemento não foi encontrado.

Pesquisa Sequencial Ordenada

- Pesquisa do valor **8**:



Binary Search

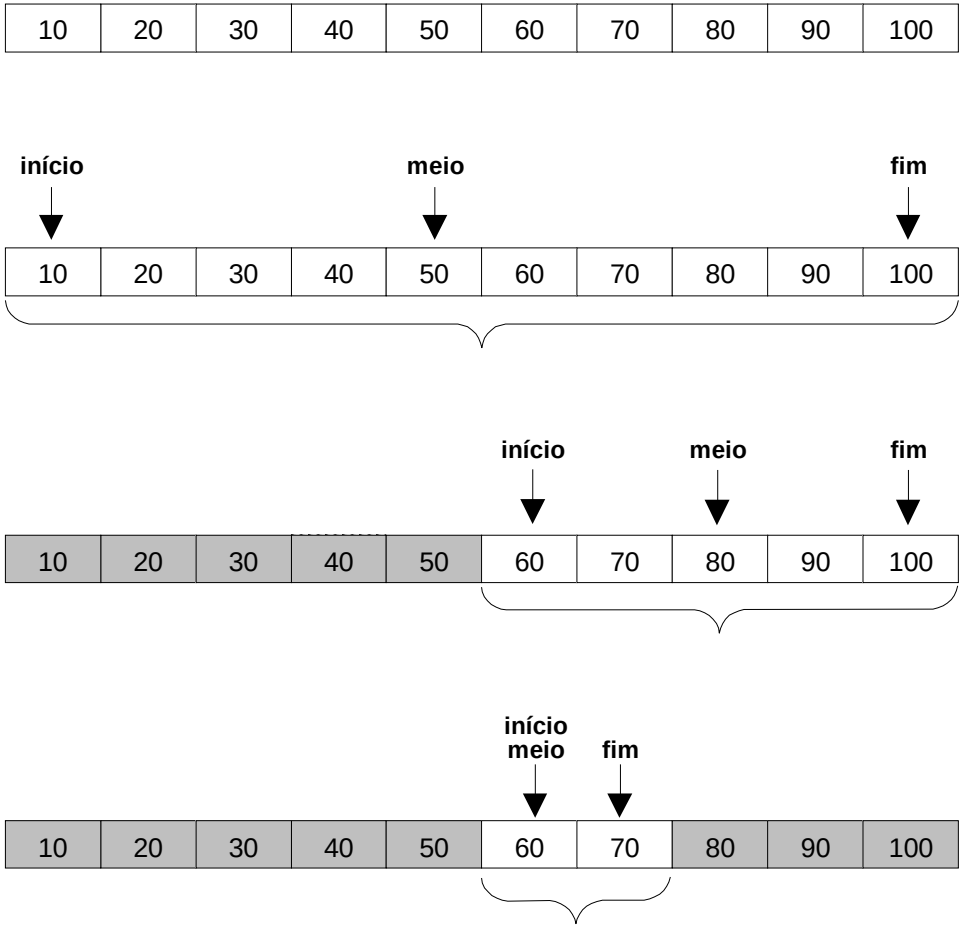
Pesquisa Binária

- Baseia-se no princípio "dividir para conquistar". A cada passo da pesquisa, o problema é dividido em dois. **Assume que os elementos do vetor estão ordenados de forma crescente.**
  - Pense no seguinte jogo: tem que acertar num número entre 1 e 20. A cada tentativa dizem-lhe se o seu palpite está acima ou abaixo do número correto. Como jogaria?
  - Provavelmente iria usar "metades" nos palpites. Exemplo:
    - 10? Menor.
    - 5? Maior.
    - 7? Menor.
    - 6? Certo!

Binary Search

Pesquisa Binária

- Pesquisa do valor **`60`**:



Ordenação

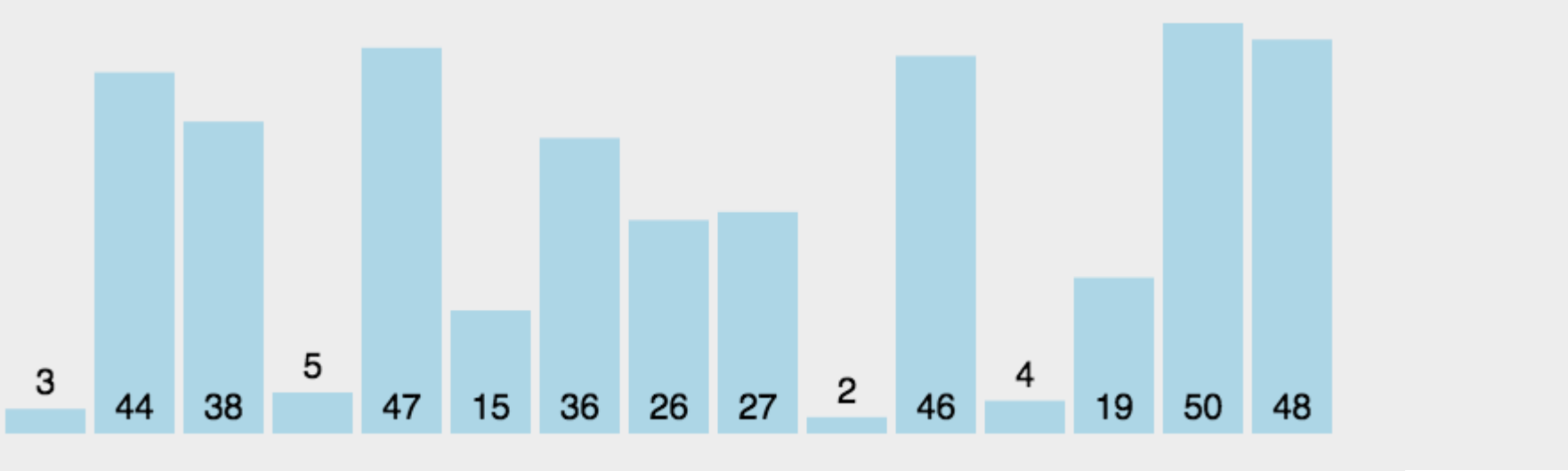
Algoritmo de ordenação

- São utilizados para ordenar coleções de dados.

Bubble Sort

Ordenação por troca

- Percorre-se sequencialmente cada par de posições adjacentes do vetor, começando na primeira posição.
- Compara-se cada par de posições adjacentes e trocam-se os elementos se não estiverem na ordem correta (se o segundo for menor do que o primeiro).
- Após cada iteração (cada vez que o vetor é percorrido), há menos um elemento (o último) que precisa de ser comparado.

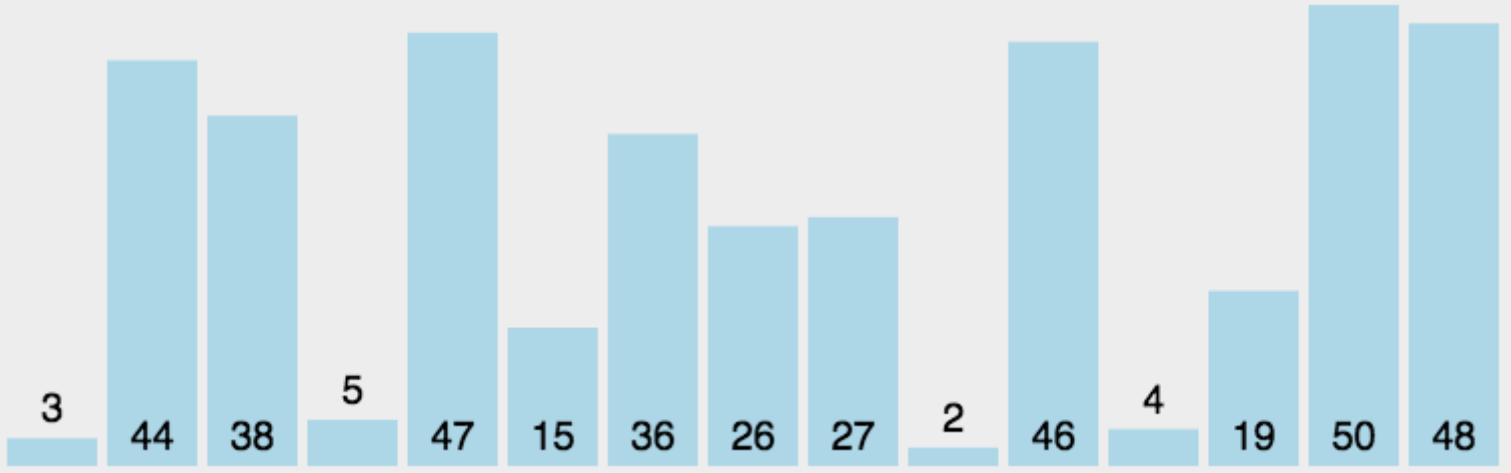


https://medium.com/@bill.shantang/8-classical-sorting-algorithms-d048eec3fdab

Selection Sort

Ordenação por seleção

- A cada passagem é selecionado o elemento de menor valor de todos os elementos analisados (no caso de se pretender uma ordenação crescente) e é colocado no sítio correto.

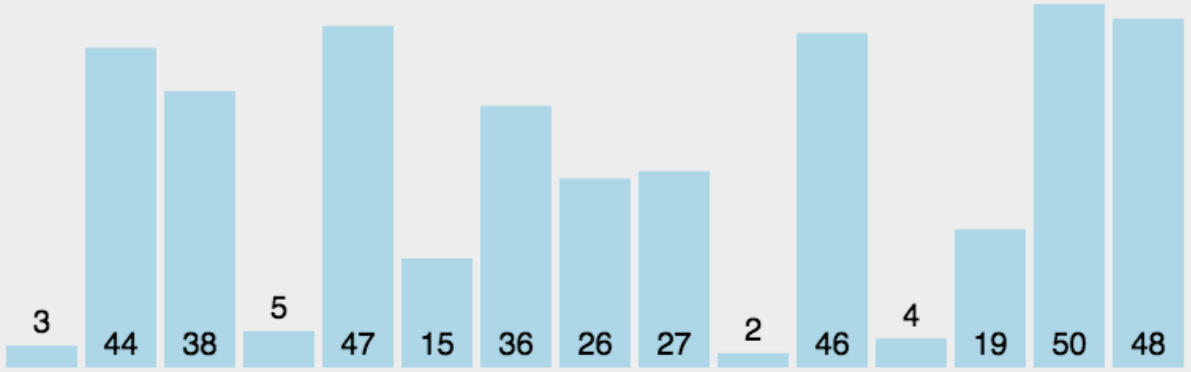


https://medium.com/@bill.shantang/8-classical-sorting-algorithms-d048eec3fdab

Insertion Sort

Ordenação por inserção

- Consiste em determinar, para cada elemento da sequência, a posição de inserção para que ele fique ordenado e inseri-lo nessa posição, deslocando para o efeito os restantes elementos.

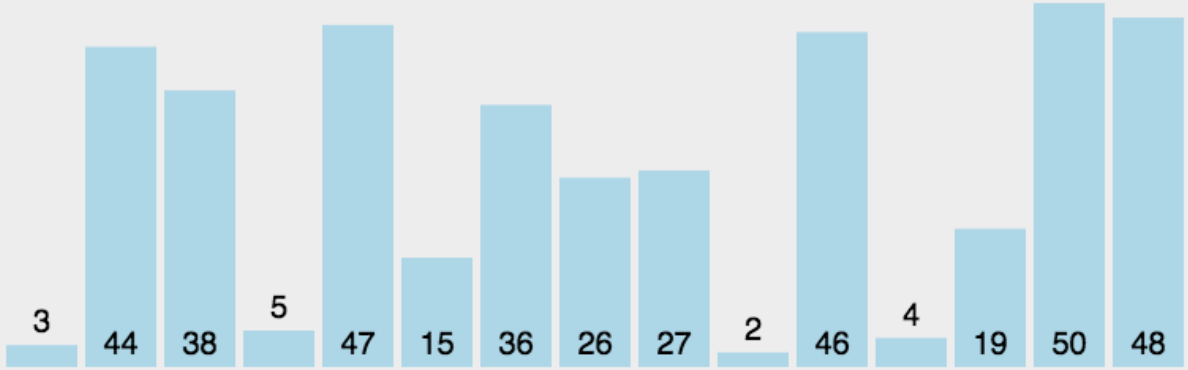


https://medium.com/@bill.shantang/8-classical-sorting-algorithms-d048eec3fdab

Merge Sort

Ordenação por fusão

- Divide o vetor em vários sub-vetores de menor dimensão, ordenando-os separadamente e depois fundindo-os.

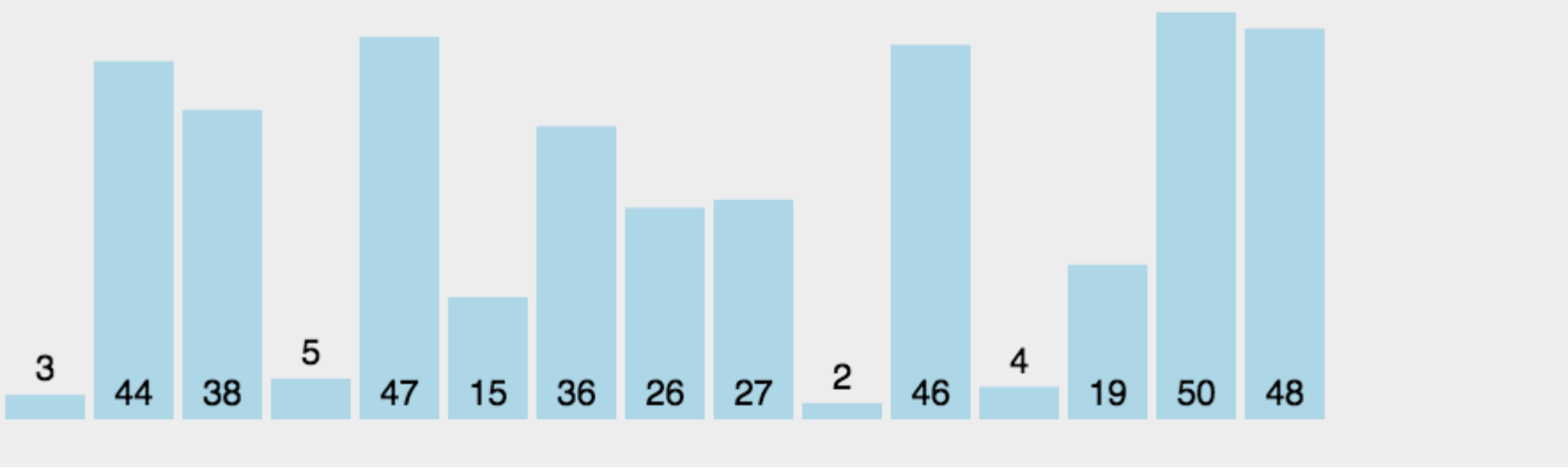


https://medium.com/@bill.shantang/8-classical-sorting-algorithms-d048eec3fdab

Quick Sort

Algoritmo de Hoare

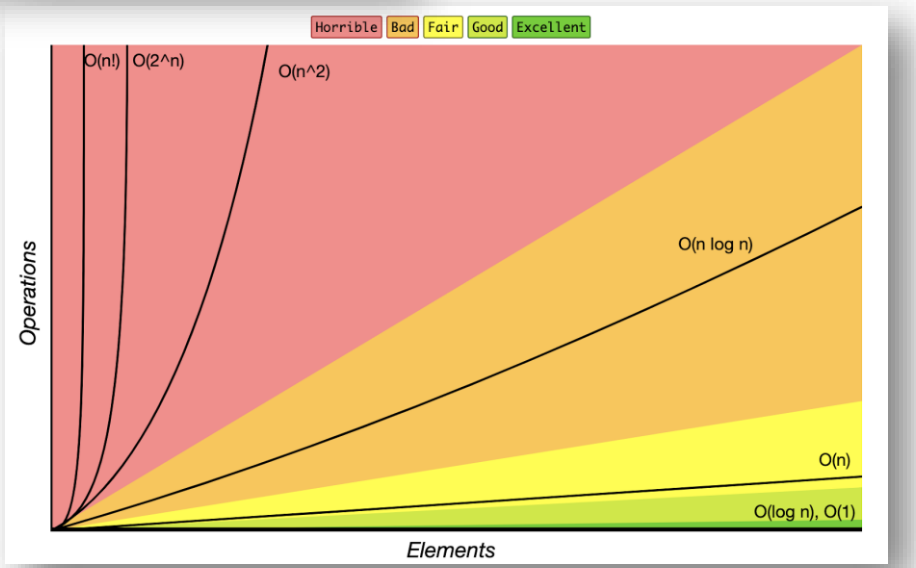
- Baseia-se no princípio "dividir para conquistar". Divide o vetor em 2 "sub-vetores".
- É escolhido um elemento do vetor, chamado elemento pivot.
- O vetor é reordenado, ficando os elementos inferiores ao pivot antes deste, e os elementos superiores ao pivot depois deste (elementos iguais podem ir para qualquer um dos lados). Após a separação, o pivot já está na sua posição final.
- Depois, é ordenado recursivamente o "sub-vetor" de elementos menores e o "sub-vetor" de elementos maiores.



https://medium.com/@bill.shantang/8-classical-sorting-algorithms-d048eec3fdab

Complexidade

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(log n)



https://www.hackerearth.com/practice/notes/sorting-and-searching-algorithms-time-complexities-cheat-sheet/
https://medium.com/@mendozabridget777/big-o-notation-time-space-complexity-9bc31952052c

Recursos recomendados

- (Capítulo 5) Rocha, A. Estruturas de Dados e Algoritmos em C; FCA – Editora de Informática, Lda, 2014; ISBN 9789727227693.
- VisuAlgo.net – visualising data structures and algorithms through animation
- Sorting Algorithms Animations
- 8 Classical Sorting Algorithms
- SORTING
- Sorting Algorithm Explained... GIF Style