

Desafios

1. Merge Sort

O Merge Sort é um algoritmo de ordenação que utiliza a estratégia de dividir para conquistar. O funcionamento consiste em dividir o array em duas metades, ordena essas metades e depois as combina o que significa que ele divide a lista original em sublistas menores, ordena cada uma delas separadamente e, em seguida, combina as sublistas ordenadas para obter a lista final ordenada.

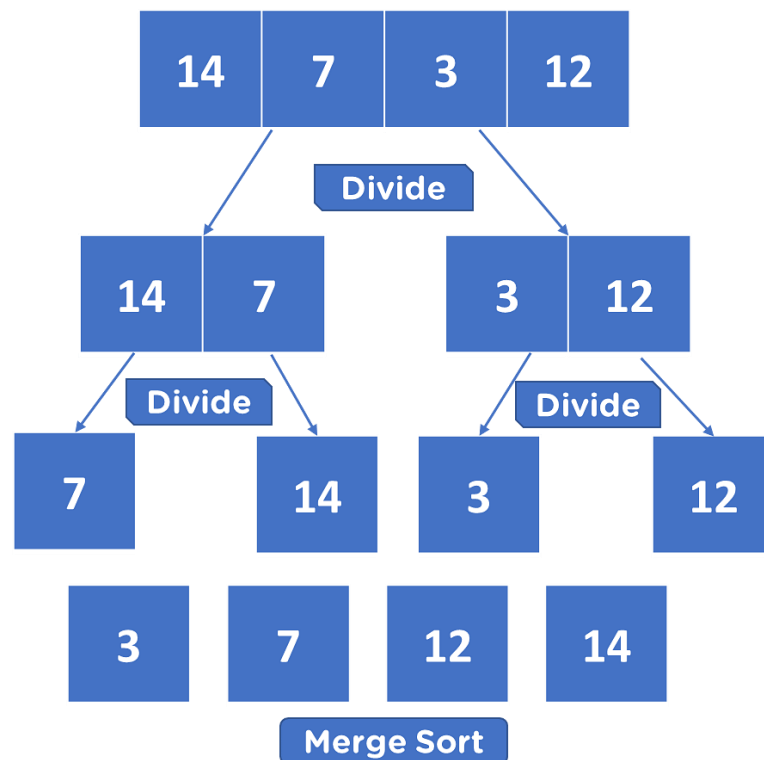


Figura 1 - <https://www.simplilearn.com/tutorials/data-structure-tutorial/merge-sort-algorithm>

2. Insertion Sort

Insertion Sort é um algoritmo que percorre um vetor de elementos da esquerda para a direita e conforme vai avançando ele ordena os elementos à esquerda.

O funcionamento do Insert Sort consiste em cada passo a partir do segundo elemento selecionar o próximo item da sequência e colocá-lo no local apropriado de acordo com o critério de ordenação.

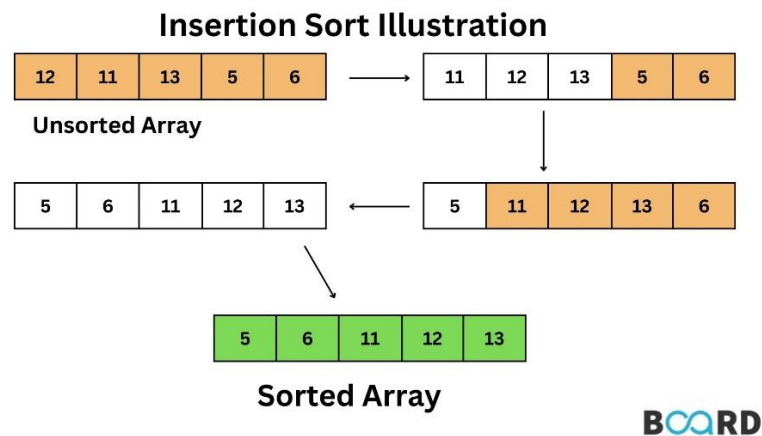


Figura 2 - <https://www.boardinfinity.com/blog/how-to-perform-insertion-sort-in-c/>

3. Bubble Sort

O bubble Sort (Ordenação por Bolhas) é um algoritmo de ordenação simples, que percorre um conjunto de elementos várias vezes, e que cada passagem, o maior elemento da sequência seja fluído para o topo. Então o Bubble Sort faz a troca de posições dos elementos até que o maior número fique por último e a lista corre em ordem crescente.

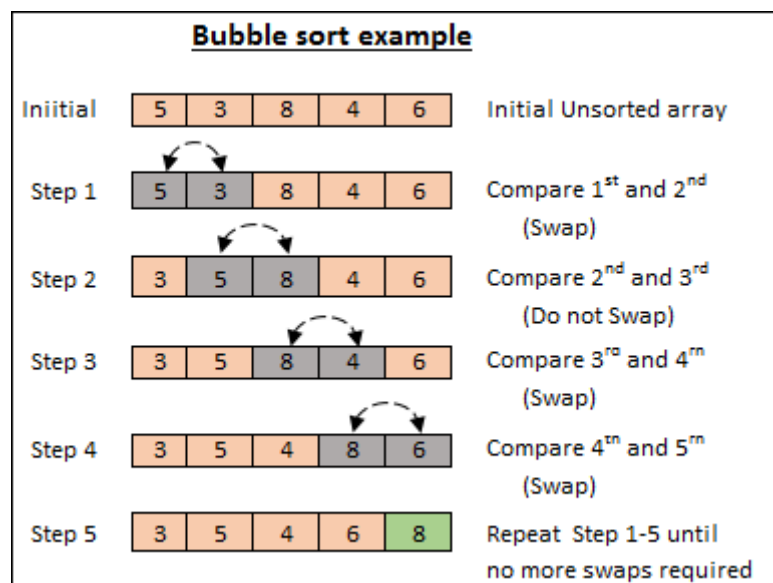


Figura 3 - <https://blog.betrybe.com/tecnologia/bubble-sort-tudo-sobre/>

4. Busca Linear

A Busca Linear é um algoritmo simples e intuitivo que localiza um valor específico em uma lista, verificando um por um. Ele começa no primeiro item, compara-o com o alvo e continua a percorrer a lista até encontrar o alvo ou chegar ao final da lista.

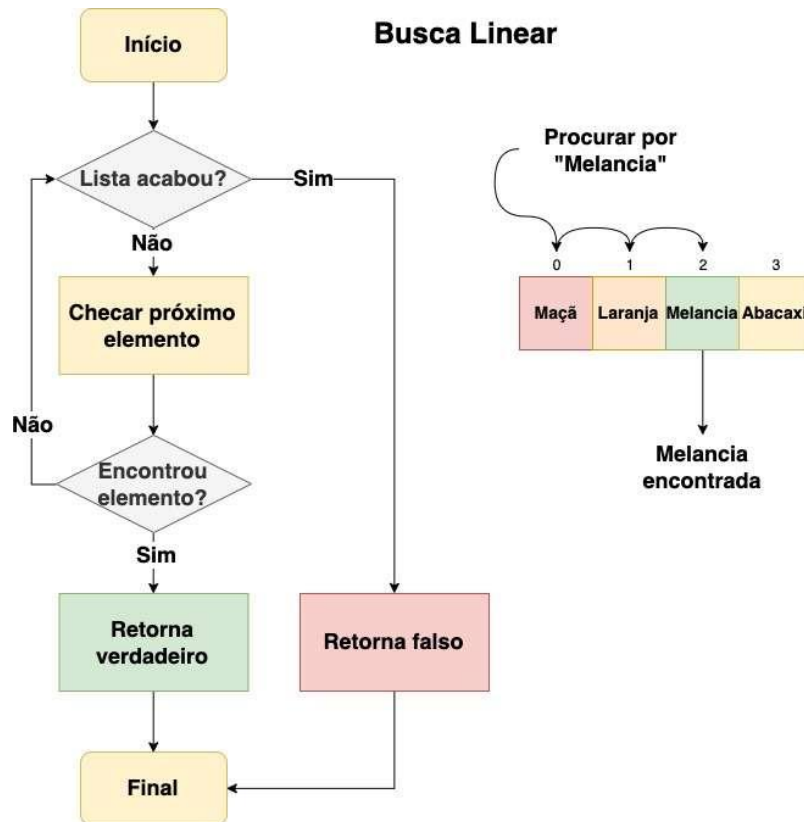


Figura 4 - <https://dicionariotec.com/posts/busca-linear>

5. Busca Binária

A busca binária é um algoritmo que encontra um item em uma lista ordenada de itens. Ela funciona dividindo repetidamente pela metade a porção da lista que deve conter o item, até reduzir as localizações possíveis a apenas uma. E diferente da Busca Linear, a Busca Binária é muito mais rápida do que a Busca Linear quando se trata de matrizes grandes.

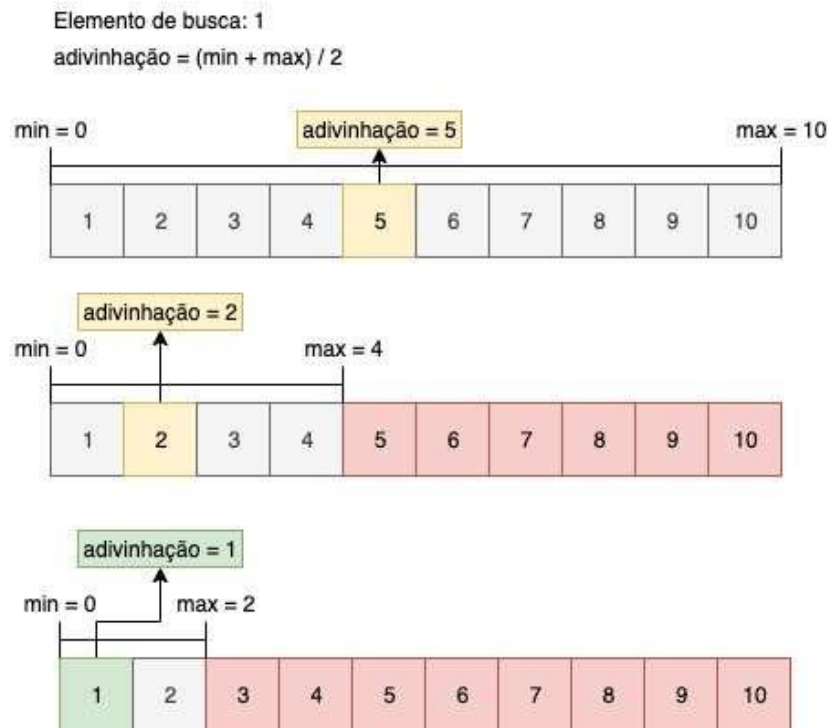


Figura 5 - <https://dicionariotec.com/posts/busca-binaria>

6. Complexidade de Algoritmo

A complexidade de um algoritmo é analisada em termos de tempo e espaço. O algoritmo pode ter um desempenho diferente dependendo do processador, disco, memória e outros parâmetros de hardware utilizados. A complexidade é usada para medir a velocidade de um algoritmo, sendo o algoritmo um agrupamento de etapas para se executar uma tarefa. Já o tempo varia de acordo com o hardware, a linguagem utilizada e o sistema operacional.

- **Merge Sort:** Tem uma complexidade de tempo $O(n \log n)$ no pior, melhor e caso médio.
- **Insertion Sort:** A complexidade do pior caso (e do caso médio) do algoritmo de ordenação por inserção é $O(n^2)$. O que significa que, no pior caso, o tempo levado para ordenar uma lista é proporcional ao quadrado do número de elementos na lista
- **Bubble Sort:** No pior caso, a lista está totalmente desordenada. Ou seja, o algoritmo precisa percorrer a lista várias vezes para ordenar todos os elementos. Em cada passagem pela lista, o algoritmo pode fazer comparações entre todos os pares de elementos e trocá-los. O número de comparações no pior caso é proporcional ao quadrado do número de elementos na lista (ou seja, $O(n^2)$), porque a cada nova passagem, o algoritmo precisa comparar quase todos os elementos.
- **Busca Linear:** A complexidade da busca linear (ou pesquisa sequencial) refere-se ao tempo necessário para encontrar um elemento

específico em uma lista ou vetor não ordenado, analisando cada elemento, um por um, até encontrar o item desejado ou chegar ao final da lista.

- **Busca Binária:** A complexidade do algoritmo é o tempo de $O(\log n)$ porque o intervalo de pesquisa é reduzido à metade a cada iteração.