**Pesquisa:**

- Marge Sort -

- Insertion Sort

- Bubble Sort

- Busca Linear/Binária

Marge Sort:

O marge sort é um algoritmo que tem o objetivo de dividir e conquistar. Ele funciona dividindo o array de entrada em subarrays menores e classificando esses subarrays e, em seguida, mesclando-os novamente para obter o array classificado.

**Dividir:**

* **[13, 15, 12, 14]** é dividido em **[13, 15** ] e **[12, 14]** .
* **[13, 15]** é dividido em **[13]** e **[15]** .
* **[12, 14]** é dividido em **[12]** e **[14]** .

**Conquistar:**

* **[13]** já está classificado.
* **[15]** já está classificado.
* **[12]** já está classificado.
* **[14]** já está classificado.

**Mesclar:**

* Mesclar **[13]** e **[15]** para obter **[13, 15]** .
* Mescle **[12]** e **[14]** para obter **[12,14]** .
* Mesclar **[13, 15]** e **[12, 14]** para obter a lista final classificada **[12, 13, 14, 15]**

Portanto, a lista ordenada é **[12, 13, 14, 15]** .

Neste sentido, o Marge sort é utilizado para dividir um algoritmo em partes menores, tornando mais claro e legível o entendimento e a resolução.

Insertion Sort

- **Insertion sort** é um algoritmo de ordenação simples no qual constrói o array ordenado final. Ele por sua vez é menos eficiente em grandes listas comparado á outros algoritmos mais avançados, porém traz vantagens como:

- Implementação simples;

- Mais eficiente na prática do que alguns dos outros algoritmos quadráticos simples em sua comparação;

- Eficiente para conjunto de dados;

- Não se altera a ordem relativa dos elementos...

O Insertion Sort aplica várias vezes a inserção ordenada para ordenar uma sequência.

**Início:** 52143

**Passo 1:** Ordenando o 2 com o 5: (25143)

**Passo 2:** Ordenando 1 com 5 e 2: (12543)

**Passo 3:** Ordenando 4 com 5: (12453)

**Passo 4:** Ordenando 2 com 1: (12534)

**Passo 5:** Ordenando 3 com 5 e 4: (12345)

**Sequência ordenada:** 12345

Bubble Sort

- **Bubble sort** é um algoritmo básico para organização de uma sequência de números ou outros elementos na ordem correta. Seu funcionamento se baseia na análise de cada conjunto de elementos adjacentes na sequência da esquerda para a direita, fazendo a alteração de sua posição se estiver incorreta. O algoritmo fica em repetição até que seja encontrar dois elementos que necessitam ser trocados.

**Início:** 62895

**Passo 1:** Ordenando 6 com 2: (26895)

**Passo 2:** Ordenando 6 com 8: (26895) (a lista permanece pois 6 < 8)

**Passo 3:** Ordenando 8 com 9: (26895) (a lista permanece pois 8 < 9)

**Passo 4:** Ordenando 9 com 5: (26859)

E esta lista de passos vai se repetindo até que a lista esteja na ordem correta (todos os elementos são reanalisados ao final da lista até ficarem ordenados corretamente)

Seguindo este exemplo a **lista ordenada** ficaria ao final: (25689)

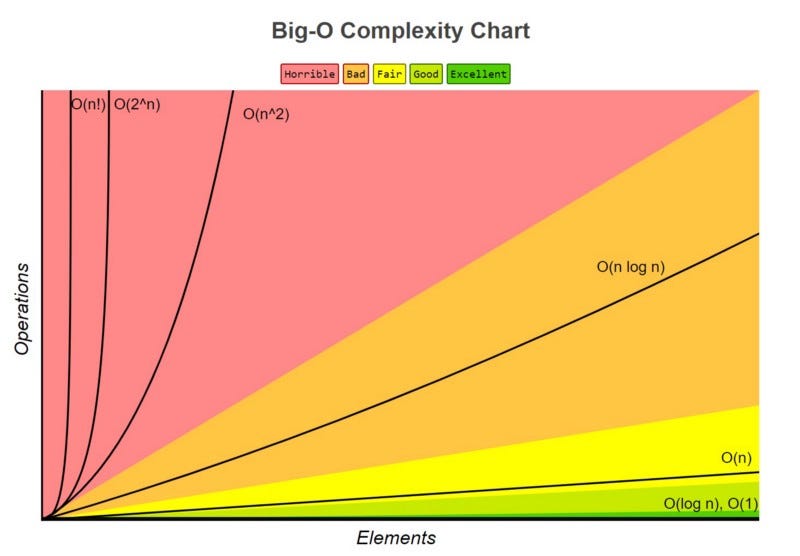
Busca Linear/Binária

**Busca Linear** o termo busca linear é utilizado para expressar um tipo de pesquisa em vetores ou listas de modo sequencial, elemento por elemento de modo que a função em relação ao tempo de elementos seja linear (cresce proporcionalmente)

**Busca binária** é um algoritmo eficiente para encontrar um item em uma lista ordenada de itens. Seu funcionamento é baseado na divisão repetida no qual segue o paradigma de divisão e conquista. Ela parte da forma como a lista está organizada e realiza sucessivas divisões comparando o elemento de busca ao elemento do meio do array, caso o elemento do meio do array seja o objetivo buscado a busca termina com sucesso. Caso o elemento venha antes do elemento de busca, então a busca segue da metade posterior do vetor.

Complexidade de algoritmo

A complexidade de algoritmo se baseia nos termos de tempo e espaço. Normalmente o algoritmo irá desempenhar um papel diferente dependendo do processador, disco, memória e outros fatores de hardware. A complexidade é utilizada para medir a velocidade de um algoritmo, sendo o algoritmo um conjunto de etapas que são utilizadas para realizar uma determinada tarefa, o tempo que leva para um algoritmo ser executado se baseia no número de passos.



**FONTES**

<https://joaoarthurbm.github.io/eda/posts/insertion-sort/>

<https://en.wikipedia.org/wiki/Insertion_sort>

<https://www.geeksforgeeks.org/merge-sort/>

<https://en.wikipedia.org/wiki/Merge_sort>

<https://pt.wikipedia.org/wiki/Bubble_sort>

<https://elemarjr.com/clube-de-estudos/artigos/o-que-e-e-como-funciona-o-bubblesort/>

<https://pt.wikipedia.org/wiki/Busca_linear>

<https://pt.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search>

<https://www.iugu.com/blog/analise-complexidade-algoritmos>