**O que é Merge Sort e como funciona?**

É um algoritmo de ordenação, como um jogo de conquistar e dividir. Ele divide uma lista até que fique apenas um número e depois começa a juntar essas divisões de forma ordenada. Este processo acontece enquanto a lista não estiver totalmente organizada.

Exemplo:

[33, 65, 44, 5, 8, 20, 3]

[33, 65, 44]  [5, 8, 20, 3]

[33, 65]  [44]  [5, 8]  [20, 3]

[33] e [65] e [44] e [5] e [8] e [20] e [3]

Até que fique:

[3, 5, 8, 20, 33, 44, 65]

**O que é Insertion Sort?**

Elabora uma lista ordenada, adicionando um elemento por vez. Ele retira os itens da lista desordenada e os coloca em suas posições na lista ordenada.

Exemplo:

[4,8,9,6,1]

[1, 4, 6, 8, 9]

**O que é Bubble Sort?**

O Bubble Sort utiliza uma lista e compara os elementos que estão juntos, verificando qual é o número com o maior valor. Depois de encontrar o maior, ele troca de posição, colocando o maior valor no final da lista. Finalizando com todos os números ordenados.

Exemplo:

[8, 3, 1,2, 5]

[1, 2, 3, 5,8]

**Busca linear:** É uma forma usada para percorrer listas. Nesse método você começa pelo primeiro item e vai analisando cada item da lista até encontrar o que está procurando ou finalizar a lista.

Exemplo: Quero encontrar o número 8.

{**1**,2,3,4,5,6,7,8}

{1,**2**,3,4,5,6,7,8}

{1,2,**3**,4,5,6,7,8}

{1,2,3,**4**,5,6,7,8}

{1,2,3,4,**5**,6,7,8}

{1,2,3,4,5,**6**,7,8}

{1,2,3,4,5,6,**7**,8}

{1,2,3,4,5,6,7,**8**}

Resumo: Em uma busca linear você precisa olhar cada item um por um.

**Busca binária:** A lista binária seria mais eficiente comparado com a linear, ela facilita a busca por itens em uma lista. Com o método de dividir para conquistar, ela divide a lista para encontrar um item específico, analisando o que não é desejado para o descarte.

Exemplo: Quero encontrar o número 8.

{1,2,3,4} --- {5,6,7,8}

{5,6} --- {7,8}

{8}

Resumo: Em uma busca binária você divide a lista e vai descartando até encontrar o que deseja.

**Pesquisa 2:**

OOP- Object Oriented Programming

O que é Herança?

A herança na programação orientada a objetos permite que classes usem propriedades e métodos de outras classes. Ou seja, ela ajuda a promover reusabilidade dos códigos, facilidade na manutenção e auxilia na organização lógica entre objetos.

Exemplo:

Classe "Maria": Tem suas próprias características, como nome, idade e cor favorita. Exemplo, sua cor favorita é verde.

Classe "Evellyn": Herda todas as características da classe “Maria”, mas tem algo a mais, como “Melhor amiga”, e sua cor favorita é azul.

Classe "Ana": Também herda tudo da “Maria”, porém tem algo que é diferente, como “Profissão”, e sua cor favorita é roxo.

Resumo: A herança facilita o código, emprestando características para outras classes.

No exemplo é possível visualizar a importância das classes utilizarem as propriedades uma das outras.

O que é Polimorfismo?

O polimorfismo é um meio na programação que permite que um objeto de diferentes classes se torne objetos de uma classe comum.

Ou seja, mesmo que os objetos sejam diferentes, eles têm o mesmo “objetivo”. Cada objeto tem características diferentes, porém todos fazem algo parecido.

Exemplo:

Maria tem várias maquiagens, todas elas têm o mesmo propósito que é ajudar a Maria a se maquiar. Porém cada item tem diferente maneiras de maquiar a Maria, como:

Blush- deixa a bochecha rosada

Corretivo- ilumina o rosto e esconde as olheiras

Batom- deixa a boca da cor desejada

Resumo: Polimorfismo é oferecer a mesma tarefa para diferentes coisas (blush, corretivo, batom) e elas fazem isso do seu próprio modo, porém você as chama do mesmo jeito(maquiar).