

COMPARAÇÃO DE CODIGOS

Estoque eficiente

CÓDIGO RECURSIVO

```
✓ def mescla_por_nome(esquerda, direita):  
    resultado = []  
    while esquerda and direita:  
        if esquerda[0].nome < direita[0].nome:  
            resultado.append(esquerda.pop(0))  
        else:  
            resultado.append(direita.pop(0))  
    resultado.extend(esquerda or direita)  
    return resultado
```

O Merge Sort é um algoritmo de ordenação baseado na técnica "Divisão e Conquista". Ele é eficiente e estável, funcionando com complexidade de tempo $O(n \log n)$, onde n é o número de elementos a serem ordenados. A ideia principal do Merge Sort é dividir a lista em partes menores, ordenar essas partes individualmente e, em seguida, mesclá-las de volta em uma lista ordenada.

CÓDIGO ITERATIVO

O laço for é usado para percorrer a lista de produtos e realizar comparações entre os itens adjacentes. O Bubble Sort é um algoritmo iterativo, pois ele repete várias vezes as comparações e trocas até que a lista esteja ordenada.

```
▼ def bubble_sort(produtos, chave):  
    n = len(produtos)  
    for i in range(n):  
  
        trocou = False  
        for j in range(0, n - i - 1):  
            if getattr(produtos[j], chave) > getattr(produtos[j + 1], chave):  
  
                produtos[j], produtos[j + 1] = produtos[j + 1], produtos[j]  
                trocou = True  
  
        if not trocou:  
            break
```

COMPARAÇÕES

Características da recursividade:

- **Estável:** Mantém a ordem relativa dos elementos iguais.
- **Complexidade:**
 - **Pior Caso, Caso Médio e Melhor Caso:** $O(n \log n)$.
- **Espaço:** O Merge Sort requer espaço adicional para armazenar as sublistas temporárias, o que resulta em complexidade espacial de $O(n)$.

Vantagens:

- Excelente desempenho mesmo em grandes listas.
- É estável, ou seja, não altera a ordem de elementos iguais.

Desvantagens:

- Consome espaço adicional $O(n)$ devido às cópias de sublistas, o que pode ser um problema em sistemas com pouca memória.

Características do Código Iterativo

- **Estabilidade:** O algoritmo é estável, mantendo a ordem de elementos iguais.
- **Complexidade:**
 - **Tempo:** O Bubble Sort tem complexidade $O(n^2)$, ineficiente para listas grandes.
 - **Espaço:** $O(1)$, já que é in-place.

Vantagens:

- Simplicidade e fácil implementação.
- Estabilidade na ordenação.
- Flexibilidade para escolher critérios de ordenação.
- Baixa utilização de memória (in-place).

Desvantagens:

- Desempenho lento com listas grandes ($O(n^2)$).
- Pouca eficiência para grandes volumes de dados.

MAIS EFICIENTE



RECURSIVIDADE

AGRADECIMENTO

AGRADECEMOS PELA ATENÇÃO