Bubble Sort:

Implemente um algoritmo eficiente para realizar a ordenação:

O Bubble Sort é um algoritmo simples de ordenação que compara repetidamente pares adjacentes de elementos e os troca se estiverem na ordem errada. Esse processo é repetido até que não ocorram mais trocas, o que indica que a lista está ordenada. Embora seja fácil de entender e implementar, o Bubble Sort é geralmente ineficiente para grandes conjuntos de dados devido à sua complexidade quadrática.

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr

arr = [64, 34, 25, 12, 22, 11, 90]
print("Antes da ordenação:", arr)
print("Após Bubble Sort:", bubble_sort(arr.copy()))
```

Execução:

```
.exe c:/Users/migue/teste/bubble.py
Antes da ordenação: [64, 34, 25, 12, 22, 11, 90]
Após Bubble Sort: [11, 12, 22, 25, 34, 64, 90]
PS C:\Users\migue\teste>
```

Documente o algoritmo utilizado e forneça uma análise da sua complexidade temporal:

Bubble Sort é um algoritmo simples de ordenação que repete repetidamente a passagem pela lista, comparando elementos adjacentes e trocando-os se estiverem na ordem errada. O processo é repetido até que a lista esteja ordenada.

Complexidade Temporal:

- Melhor caso:O(n) quando a lista já está ordenada (ocorre apenas uma passagem).
- Caso médio: O(n^2).
- Pior caso: O(n^2) quando a lista está ordenada na ordem inversa.