

## 18. Ordenação Estável e Instável

- Identifique quais algoritmos de ordenação da lista são estáveis e explique o que isso significa. Demonstre com exemplos.

**Estáveis: Merge Sort, Radix Sort, Bucket Sort**

Um algoritmo de ordenação é considerado estável se preserva a ordem de registros de chaves iguais. Isso significa que, após a ordenação, dois elementos equivalentes permanecem na mesma ordem em que estavam antes da ordenação. Lista: [3a, 1, 3b, 2]. Após ordenação: [1, 2, 3a, 3b], note que os elementos 3a e 3b mantiveram sua ordem relativa após a ordenação.

Exemplo:

```
lista = [62, 12, 62, 98, 10, 24, 11, 67]
resultado = bucket_sort(lista)
print("Lista ordenada:", resultado)
```

```
Lista ordenada: [10, 11, 12, 24, 62, 62, 67, 98]
```

```
numeros = [38, 27, 43, 3, 9, 82, 10, 3]
resultado = merge_sort(numeros)
print(f"Lista ordenada: {resultado}")
```

```
Lista ordenada: [3, 3, 9, 10, 27, 38, 43, 82]
```

```
num2dig = [170, 45, 75, 90, 802, 24, 2, 66, 66]
num5dig = [12345, 54321, 67890, 98765, 11111]
num10dig = [9876543210, 1234567890, 1122334455, 9988776655]
ordem = radix_sort(num2dig)
print("Lista ordenada: \n", ordem)
```

```
Lista ordenada:
[2, 24, 45, 66, 66, 75, 90, 170, 802]
```

```
Lista ordenada:
```