

## Tutoria 10

### Questão 1.

#### COUNTING SORT:

Vetor Original A: [9, 2, 3, 11, 4, 6, 1, 8]

Maior valor (k): 11

Passo a Passo:

- Contagem: Frequência de cada elemento em um vetor auxiliar C[0..11].  
 $C = [0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1]$
- Acúmulo: Soma prefixada para determinar posições.  
 $C = [0, 1, 2, 3, 4, 4, 5, 5, 6, 7, 7, 8]$
- Posicionamento: O elemento A[i] vai para a posição C[A[i]]-1 no vetor B.

Resultado: [1, 2, 3, 4, 6, 8, 9, 11]

#### RADIX SORT:

Vetor Original A: [126, 789, 213, 456, 481, 510, 471, 900]

Passo a Passo:

- Ordenação por Unidades ( $10^0$ ):  
[510, 900, 481, 471, 213, 126, 456, 789]
- Ordenação por Dezenas ( $10^1$ ):  
[900, 510, 213, 126, 456, 471, 481, 789]
- Ordenação por Centenas ( $10^2$ ):  
[126, 213, 456, 471, 481, 510, 789, 900]

Resultado: [126, 213, 456, 471, 481, 510, 789, 900]

#### BUCKET SORT:

Vetor Original A: [0.13, 0.11, 0.34, 0.54, 0.90, 0.76, 0.32, 0.21]

Número de baldes: 10 (Índices 0-9)

Passo a Passo:

- Distribuição (Posição = valor \* 10):  
B1: [0.13, 0.11]  
B2: [0.21]  
B3: [0.34, 0.32]  
B5: [0.54]  
B7: [0.76]  
B9: [0.90]
- Ordenação Interna:  
B1: [0.11, 0.13]  
B3: [0.32, 0.34]
- Concatenação:  
[0.11, 0.13, 0.21, 0.32, 0.34, 0.54, 0.76, 0.90]

Resultado: [0.11, 0.13, 0.21, 0.32, 0.34, 0.54, 0.76, 0.90]

## **Questão 2.**

### **BUSCA SEQUENCIAL (OU LINEAR)**

- Funcionamento: Percorre o vetor elemento por elemento, do início ao fim, comparando cada posição com o valor buscado.
- Comportamento: Simples, mas ineficiente para grandes volumes de dados.
- Complexidade:  $O(n)$ .

### **BUSCA BINÁRIA**

- Funcionamento: Utiliza a estratégia de "divisão e conquista". Compara o valor buscado com o elemento central do vetor.
  - Se o valor for igual ao meio, a busca termina.
  - Se for menor, repete a busca na metade esquerda.
  - Se for maior, repete a busca na metade direita.
- Comportamento: Extremamente rápida, pois reduz o espaço de busca pela metade a cada passo.
- Complexidade:  $O(\log n)$ .

### **COMPARATIVO:**

- Busca Sequencial: Funciona em qualquer vetor; custo  $O(n)$ .
- Busca Binária: Exige ordenação prévia; custo  $O(\log n)$ .

## **Questão 4.**

Vetor A = [1, 4, 10, 13, 19, 21, 29, 30, 32, 43, 67, 76, 78, 91, 93, 99]

Sim, é possível realizar a busca binária pois o vetor está ordenado.

Passo a Passo:

1. Intervalo [0..15], Meio = 7 (Valor 30).  $4 < 30$ , busca na esquerda.
2. Intervalo [0..6], Meio = 3 (Valor 13).  $4 < 13$ , busca na esquerda.
3. Intervalo [0..2], Meio = 1 (Valor 4).  $4 == 4$ , ENCONTRADO.

Resultado: Sucesso, índice 1.