

# Contagem de ocorrências em uma sequência

Alisson  
Denis  
Fernanda

# O PROBLEMA

- › Determinar, para cada elemento, o número de ocorrências em uma sequência

# SOLUÇÃO TRIVIAL

- › Uma solução de força-bruta é comparar cada elemento específico com todos os elementos da sequência.
- › Existem  $n^2$  pares de elementos para serem comparados

CONTAGEM-OCORRÊNCIAS( $A, n$ )

**para**  $i \leftarrow 1$  **até**  $n$  **faça**

**para**  $j \leftarrow 1$  **até**  $n$  **faça**

**se**  $A[i] = A[j]$

**então**  $\text{contar} \leftarrow \text{contar} + 1$

# SOLUÇÃO TRIVIAL

- › Demonstrar a ordem de complexidade

# SOLUÇÃO TRIVIAL

› Trecho do código responsável pelo tempo exponencial

```
1 void executaOperacaoTrivial(int *vetor, int N) {  
2     int i, j, contar = 0;  
3  
4     for (i=0; i<N; i++) {  
5         for (j=0; j<N; j++)  
6             if (vetor[i] == vetor[j])  
7                 contar++;  
8         contar = 0;  
9     }  
10  
11     return;  
12 }
```

# SOLUÇÃO *DIVIDE AND CONQUER*

- › Reduzir a um problema conhecido: ordenação
- › É possível ordenar em tempo  $O(n \log n)$  utilizando a técnica de **divisão e conquista**
- › Listar as opções de algoritmos conhecidos e suas ordens de complexidade e exibir a escolha pelo mergesort
- › Falar sobre essa técnica
- › Como a ordenação requer tempo  $O(n \lg n)$  e o cálculo após a ordenação requer tempo  $O(n)$ , esta solução usa o tempo  $O(n \lg n)$ .

# **SOLUÇÃO *DIVIDE AND CONQUER***

- › Incluir pseudocódigo

# **SOLUÇÃO *DIVIDE AND CONQUER***

- › Demonstrar a ordem de complexidade:
  - Demonstrar a árvore de recursão (incluir imagem)
  - Demonstrar pelo método da substituição



# **SOLUÇÃO *DIVIDE AND CONQUER***

› Exibir trecho do código

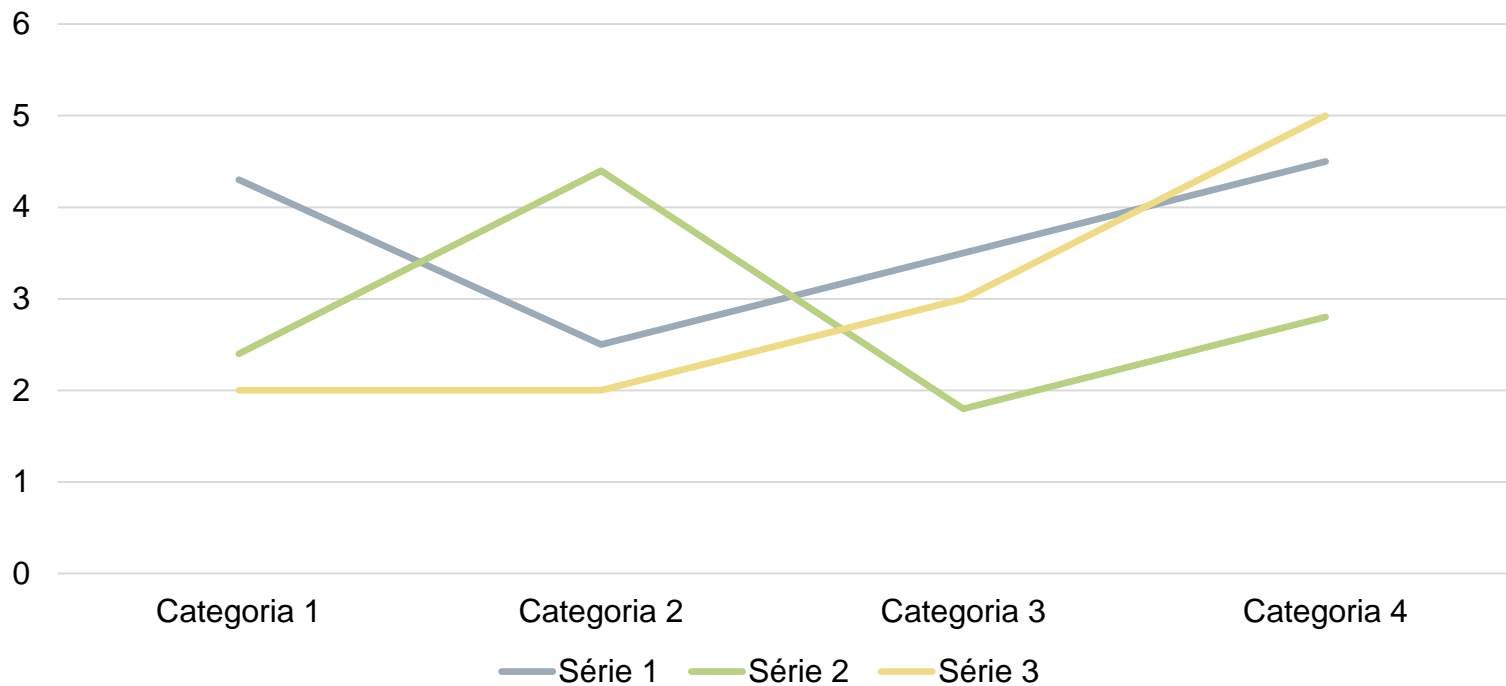
# COMPARAÇÃO DAS ABORDAGENS

› Tabelas com destaques de tempo

N	Alg A	Alg B
1000	82	95
10000	76	88
100000	84	90

# COMPARAÇÃO DAS ABORDAGENS

› Gráfico de curvas



# ALGORITMO $O(n)$

- › Verificar se tem essa solução