

# Algoritmos de ordenação

## Instruções para o Usuário

Este programa implementa quatro algoritmos de ordenação clássicos: **Bubble Sort**, **Insertion Sort**, **Merge Sort** e **Quick Sort**. A seguir, estão as instruções detalhadas sobre como usar a aplicação.

### 1. Compilação do Código

Primeiro, é necessário compilar o código em C para gerar o executável. Se você ainda não tem o código compilado, siga estas etapas:

- **Pré-requisitos:** Certifique-se de ter um compilador C instalado, como o `gcc` (GNU Compiler Collection). Para sistemas Linux ou macOS, o `gcc` pode ser instalado via terminal. No Windows, você pode usar o MinGW ou outro compilador C.
- **Compilando o Código:**
  - Salve o código fornecido em um arquivo com a extensão `.c` (por exemplo, `ordenacao.c`).
  - No terminal, navegue até o diretório onde o arquivo está salvo.
  - Use o comando abaixo para compilar o código: `gcc -o ordenacao ordenacao.c`  
Isso gerará um executável chamado `ordenacao` (ou `ordenacao.exe` no Windows).

### 2. Executando o Programa

Depois de compilar o código, execute o programa para ver os resultados da ordenação dos arrays.

- **Executando no Terminal:**
  - No Linux ou macOS: `./ordenacao`
  - No Windows: `ordenacao.exe`

O programa será executado e exibirá o estado dos arrays antes e depois da aplicação de cada algoritmo de ordenação.

### 3. Alterando os Dados de Entrada

O programa foi projetado para trabalhar com arrays definidos diretamente no código-fonte. Para testar diferentes dados de entrada (ou seja, diferentes arrays), você pode editar os valores dos arrays no código.

- **Como modificar os arrays:**

- Abra o arquivo fonte (por exemplo, `ordenacao.c`) em um editor de texto.
- Localize a parte do código onde os arrays são definidos, como no exemplo abaixo:

```
int arr1[] = {64, 25, 12, 22, 11};  
int arr2[] = {64, 34, 25, 12, 22, 11};  
int arr3[] = {12, 11, 13, 5, 6, 7};  
int arr4[] = {10, 7, 8, 9, 1, 5};
```

- Modifique os números dentro de cada array conforme desejado.  
Exemplo:

```
int arr1[] = {5, 3, 8, 1, 4};
```

- Após realizar as modificações, salve o arquivo.

### 4. Visualizando a Saída

Ao executar o programa, ele imprimirá no terminal o estado de cada array antes e depois de ser ordenado por cada algoritmo.

**Exemplo de Saída:**

Array antes de ordenar (Bubble Sort):

64 25 12 22 11

Array após ordenar (Bubble Sort):

11 12 22 25 64

Array antes de ordenar (Insertion Sort):

64 34 25 12 22 11

Array após ordenar (Insertion Sort):

11 12 22 25 34 64

Array antes de ordenar (Merge Sort):

12 11 13 5 6 7

Array após ordenar (Merge Sort):

5 6 7 11 12 13

Array antes de ordenar (Quick Sort):

10 7 8 9 1 5

Array após ordenar (Quick Sort):

1 5 7 8 9 10

## 5. Interação com o Programa

- **Alteração dos Arrays:** Caso você deseje testar diferentes conjuntos de números, basta alterar os arrays no código.
- **Escolha dos Algoritmos:** O programa já executa todos os algoritmos de ordenação (Bubble Sort, Insertion Sort, Merge Sort e Quick Sort). Cada um será executado sequencialmente sobre os arrays.

## Resumo

- Compile o código com o comando `gcc -o ordenacao ordenacao.c`.
- Execute o programa gerado (`./ordenacao` ou `ordenacao.exe`).
- Edite os arrays no código para testar diferentes entradas.
- Veja os resultados da ordenação no terminal.

Com essas instruções, o usuário pode facilmente compilar, executar e modificar os dados para testar os algoritmos de ordenação.

## Algoritmo dos conjuntos disjuntos

### Instruções para o Usuário

Este programa permite trabalhar com **conjuntos disjuntos** usando a estrutura de dados "Union-Find" com **compressão de caminho** e **união por tamanho**. O programa

lê dados de entrada de um arquivo, processa a união de subconjuntos e imprime os conjuntos disjuntos resultantes.

## Como Usar o Programa:

### 1. Compilar o Código:

- a. Para compilar o código C, você pode usar um compilador como o gcc. Execute o seguinte comando no terminal: `gcc -o conjuntos_disjuntos programa.c`  
Onde `programa.c` é o nome do arquivo que contém o código-fonte.

### 2. Executar o Programa:

- a. Após compilar o programa, você pode executá-lo da seguinte forma:  
`./conjuntos_disjuntos`
- b. O programa irá ler os dados de entrada do arquivo `input.txt` (veja o formato abaixo), processará os subconjuntos e imprimirá os conjuntos disjuntos resultantes.

### 3. Formato do Arquivo de Entrada (`input.txt`):

- a. O arquivo de entrada deve estar no formato abaixo:

```
n k
s1 e1 e2 ... es1
s2 e1 e2 ... es2
...
sk e1 e2 ... esk
```

Onde:

- b. **n**: O número de elementos (inteiros de 1 a n).
- c. **k**: O número de subconjuntos.
- d. Cada linha subsequente descreve um subconjunto, começando com um número `si`, que é o número de elementos no subconjunto, seguido pelos `si` elementos desse subconjunto.

### Exemplo de Arquivo de Entrada:

```
7 3
3 1 2 3
2 4 5
3 6 7 3
```

Neste exemplo:

- e. Existem 7 elementos (de 1 a 7).
- f. Existem 3 subconjuntos:
  - i. O primeiro subconjunto contém os elementos {1, 2, 3}.
  - ii. O segundo subconjunto contém os elementos {4, 5}.
  - iii. O terceiro subconjunto contém os elementos {6, 7, 3}.

O programa irá unir os elementos conforme descrito nos subconjuntos e, ao final, exibirá os conjuntos disjuntos resultantes.

#### 4. Saída:

- a. O programa imprimirá os conjuntos disjuntos após processar as uniões.  
Para o exemplo acima, a saída será algo como:

Conjunto: 1 2 3 4 5 6 7

Isso indica que todos os elementos foram unidos em um único conjunto, já que o elemento 3 apareceu em dois subconjuntos e conectou todos os outros elementos.

### Observações:

- **Entrada de Arquivo:** Certifique-se de que o arquivo `input.txt` esteja no mesmo diretório que o programa ou forneça o caminho completo.
- **Modificação do Arquivo de Entrada:** Se necessário, modifique o arquivo de entrada para refletir diferentes números de elementos ou subconjuntos, adaptando o número `n` de elementos e a descrição de cada subconjunto.

Se houver qualquer dúvida ou erro ao executar o programa, verifique o formato do arquivo e certifique-se de que o código foi compilado corretamente.