

Instruções Detalhadas para Compilar e Executar o Código

Para compilar e executar o projeto de algoritmos de ordenação, você precisará de um ambiente de desenvolvimento C/C++ com o compilador GCC e o utilitário `make` instalados. Siga os passos abaixo:

Pré-requisitos

Certifique-se de ter os seguintes softwares instalados em seu sistema:

- **GCC (GNU Compiler Collection):** O compilador C.
- **Make:** Um utilitário para gerenciar a compilação de projetos.

Instalação no Linux (Ubuntu/Debian)

Abra um terminal e execute os seguintes comandos:

```
sudo apt update  
sudo apt install build-essential
```

Este comando instalará o GCC, G++ e o `make`.

Instalação no macOS

Instale as ferramentas de linha de comando do Xcode:

```
xcode-select --install
```

Isso instalará o GCC e o `make`.

Instalação no Windows

Recomenda-se usar o [MinGW-w64](#) ou o [WSL \(Windows Subsystem for Linux\)](#). Se optar pelo MinGW-w64, certifique-se de adicionar o diretório `bin` do MinGW-w64 ao seu PATH do sistema.

Passos para Compilar e Executar

1. **Organize os Arquivos:** Certifique-se de que todos os arquivos do projeto (`main.c`, `bubble.c`, `shell.c`, `quick.c`, `insertion.c`, `selection.c`, `utils.c`, `sort.h`, `utils.h`, `Makefile`) estejam no mesmo diretório em seu computador.
2. **Abra o Terminal/Prompt de Comando:** Navegue até o diretório onde você salvou os arquivos do projeto usando o comando `cd`.

Exemplo: `bash cd /caminho/para/seus/arquivos/do/projeto`

3. **Compile o Projeto:** No terminal, execute o comando `make`. O `Makefile` configurado irá automatizar o processo de compilação, criando os arquivos objeto (`.o`) e, finalmente, o executável `sort_test`.

```
bash make
```

Se a compilação for bem-sucedida, você não verá mensagens de erro e um arquivo executável chamado `sort_test` será criado no mesmo diretório.

- **Solução de Problemas de Compilação:**

- Se você receber erros como `gcc: command not found` ou `make: command not found`, verifique se os pré-requisitos (GCC e Make) estão corretamente instalados e configurados no PATH do seu sistema.
- Qualquer outro erro de compilação indicará um problema no código-fonte. Revise as mensagens de erro para identificar a causa.

4. **Execute o Programa:** Após a compilação bem-sucedida, execute o programa no terminal usando o seguinte comando:

```
bash ./sort_test
```

O programa irá então:

- * Gerar vetores de números inteiros aleatórios com tamanhos de 1.000, 10.000 e 100.000 elementos.
- * Aplicar os algoritmos Bubble Sort, Shell Sort, Quick Sort, Insertion Sort e Selection Sort a cópias desses vetores.
- * Medir e exibir o tempo de execução de cada algoritmo para cada tamanho de vetor.
- * Indicar se o vetor foi ordenado corretamente (`(OK)`) ou se houve algum erro (`(Erro)`).

A saída no terminal será semelhante a esta: `` `==== Teste de Algoritmos de Ordenação====

Tamanho do vetor: 1000 Bubble Sort: 0.0008 s (OK) Shell Sort: 0.0001 s (OK)
Quick Sort: 0.0001 s (OK) Insertion Sort:0.0001 s (OK) Selection Sort:0.0004 s (OK)

Tamanho do vetor: 10000 Bubble Sort: 0.1235 s (OK) Shell Sort: 0.0010 s (OK)
Quick Sort: 0.0006 s (OK) Insertion Sort:0.0120 s (OK) Selection Sort:0.0323 s (OK)

Tamanho do vetor: 100000 Bubble Sort: 15.7291 s (OK) Shell Sort: 0.0142 s (OK)
Quick Sort: 0.0075 s (OK) Insertion Sort:1.1906 s (OK) Selection Sort:3.3544 s (OK)
`` `

5. **Limpar Arquivos Compilados (Opcional):** Se desejar remover os arquivos objeto (`.o`) e o executável `sort_test` gerados pela compilação, você pode usar o comando `make clean` :

`bash make clean` Isso é útil para garantir uma compilação "limpa" se você fizer grandes alterações no código ou no `Makefile` .