

Gnome Sort



1. Introdução

O projeto GnomeSortClientes é um software desenvolvido para demonstrar a implementação e a análise do algoritmo de ordenação Gnome Sort aplicado à ordenação de clientes por seus nomes. O objetivo do projeto é fornecer uma compreensão clara do funcionamento do algoritmo Gnome Sort e sua eficiência na ordenação de dados.

2. Funcionalidades:

- Implementação do algoritmo Gnome Sort em linguagem C para ordenar uma lista de clientes por seus nomes.
- Arquivos organizados: o projeto é composto por três arquivos principais, **main.c**, **cliente.c** e **cliente.h**, que contêm a implementação do algoritmo, a estrutura de dados do cliente e as definições de função, respectivamente.
- Visualização de resultados: o programa exibe os resultados da ordenação na saída padrão.

3. Funcionamento do Algoritmo:

O algoritmo Gnome Sort é uma variação simples do algoritmo de ordenação por inserção. Ele percorre repetidamente a lista de elementos, comparando cada elemento com o anterior. Se os elementos estiverem em ordem, avança para o próximo elemento; caso contrário, troca os elementos de lugar e retrocede na lista. Esse processo continua até que toda a lista esteja ordenada.

4. Organização do Projeto:

- **main.c:** Contém a função principal do programa, que chama a função de ordenação Gnome Sort.
- **cliente.c:** Define a estrutura de dados do cliente e implementa a função de ordenação Gnome Sort.
- **cliente.h:** Declaração das estruturas de dados e protótipo de funções relacionadas aos clientes.
- **README.md:** Documentação do projeto, incluindo uma explicação detalhada do algoritmo Gnome Sort, instruções para compilar e executar o programa, análise de complexidade e referências.

5. Compilação e Execução:

Para compilar o projeto, execute o seguinte comando no terminal:

```
gcc -o main main.c cliente.c
```

Em seguida, para executar o programa compilado, use o comando:

```
./main
```

6. Análise de Complexidade:

O algoritmo Gnome Sort apresenta as seguintes complexidades de tempo:

- **Melhor Caso:** $O(n)$, onde n é o número de elementos no array. Neste caso, o array já está ordenado, e o algoritmo percorre o array uma vez sem precisar fazer trocas.
- **Pior Caso:** $O(n^2)$, onde n é o número de elementos no array. Neste caso, o array está ordenado inversamente, e o algoritmo precisa fazer o máximo de trocas em cada iteração.

A finalidade de computar o tempo de execução é medir o desempenho do algoritmo Gnome Sort em diferentes situações, permitindo comparar seu desempenho com outros algoritmos de ordenação e avaliar sua eficiência em conjuntos de dados de diferentes tamanhos.

7. Conclusão

O projeto GnomeSortClientes fornece uma implementação prática do algoritmo Gnome Sort, permitindo uma compreensão clara de seu funcionamento e eficiência. Embora seja um algoritmo simples, o Gnome Sort pode ser útil em certos contextos e serve como um excelente exercício de aprendizado.