

Trabalho Prático 03 – AEDS 1 – Trios

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega (código + implementação): 27/01/2025

Data das Entrevistas: 28/01 e 30/01 (nas aulas de laboratório)

Forma de Entrega: PVANet (formato .zip ou .tar.gz)

Ordenação de Rochas Minerais

Objetivo:

Desenvolver um programa em C que utiliza algoritmos de ordenação para organizar uma lista de Rochas Minerais desordenadas. A implementação deve explorar conceitos de TADs (Tipos Abstratos de Dados) e eficiência computacional.

Descrição do Trabalho

1. Reimplementação dos TADs:

- Utilize o TAD **Rocha Mineral** para representar as informações básicas de uma rocha, conforme o TP1.
- Reimplemente o TAD **Compartimento**, utilizando um vetor para armazenar e manipular a coleção de Rochas Minerais.

2. Algoritmos de Ordenação:

- Inclua no TAD **Compartimento** duas operações de ordenação:
 - Um algoritmo simples (à escolha do grupo), como BubbleSort ou SelectionSort.
 - Um algoritmo mais sofisticado (à escolha do grupo), como ShellSort ou QuickSort.
- Ambas as operações devem permitir a ordenação das Rochas por peso.

3. Métricas de Desempenho:

- Para cada algoritmo de ordenação, registre:
 - Número de comparações realizadas.
 - Número de movimentações de itens no vetor.

- Tempo total de execução.

4. **Entrada e Tamanhos dos Compartimentos:**

- O sistema deverá processar compartimentos de diferentes tamanhos: 250, 1000 e 10000 Rochas.
- Uma lista de Rochas Minerais será fornecida por entrada de arquivo.

5. **Saída:**

- A saída será representada por uma lista ordenada onde cada linha deve conter o nome e o peso da rocha. Por fim, na última linha deve-se apresentar as métricas de execução, juntamente com o nome do algoritmo usado.

Análise e Relatório Final

- Compare os resultados obtidos pelos dois algoritmos implementados.
- Discuta as vantagens e desvantagens de cada abordagem e situações em que seriam mais adequadas.
- Produza um relatório final contendo:
 1. Introdução ao problema e objetivos.
 2. Descrição dos algoritmos de ordenação e suas implementações.
 3. Resultados dos testes realizados.
 4. Discussão das conclusões.
 5. Conclusões gerais sobre a aplicação dos algoritmos no problema.

CrITÉrios de Avaliação

1. Implementação correta dos TADs e algoritmos.
2. Precisão na ordenação e consistência dos resultados.
3. Análise de desempenho detalhada.
4. Organização e clareza do relatório final.

Arquivo de Entrada

Explicação:

- **Primeira linha:** Contém o valor 10, que indica o número total de rochas (N).

- **Linhas subsequentes:** Cada linha contém a latitude, longitude, peso e uma sequência de até três minerais, separados por um espaço.

```
10
76.926237 147.437399 12.5 Ferrolita
-28.452065 84.537518 8.2 Solarium
55.489421 -150.457196 15.3 Aquavitae Terranita
9.344208 172.900976 9.8 Terranita Calaris
-150.457196 76.926237 7.1 Terranita Solarium
55.489421 147.437399 14 Aquavitae Terranita
-15.677243 -44.787665 11.4 Calaris Aquavitae
71.602894 -172.649438 10.2 Solarium Ferrolita
80.71773 -27.109002 13.6 Ferrolita Terranita
71.602894 -172.649438 6.9 Calaris Aquavitae Ferrolita
```

Arquivo de Saída

Explicação:

1. A lista foi ordenada em ordem crescente de peso.
2. As métricas de execução foram calculadas:
 - **Comparações:** Número de comparações realizadas pelo algoritmo (exemplo: 45).
 - **Movimentações:** Quantidade de trocas feitas entre elementos do vetor (exemplo: 36).
 - **Tempo de execução:** Tempo total gasto na ordenação (exemplo: 0.002 segundos).
 - **Algoritmo:** Nome do algoritmo usado para realizar a ordenação (neste caso, **BubbleSort**).

```
Aquacalis 6.9
Terrasol 7.1
Solaris 8.2
Terrolis 9.8
Solarisfer 10.2
Calquer 11.4
Ferrom 12.5
Terralis 13.6
Aquaterra 14.0
Aquaterra 15.3
```

```
Comparações: 45
Movimentações: 26
Tempo de execução: 0.0000s
Algoritmo: BubbleSort
```