

Universidade Federal do Rio de Janeiro

Relatório do Desenvolvimento de Estruturas de Dados Lineares

Charles Ribeiro Chaves - 122086950

Filipe Viana da Silva - 121050053

Vinícius Brasil de Oliveira Barreto - 120029237

Sumário

1	Introdução	3
2	Estruturas de Dados Utilizadas	3
2.1	Classe Abstrata Base	3
2.2	IndexedArray	3
2.3	Lista Encadeada Simples e Dupla	3
2.4	Pilha e Fila	3
2.5	Lista Circular Simples	3
2.6	Fila de Prioridade	3
3	Divisão de Módulos	3
4	Descrição das Rotinas e Funções	4
5	Complexidades de Tempo e Espaço	4
6	Problemas e Observações	4
7	Conclusão	4

Resumo

Este relatório apresenta o desenvolvimento de diversas estruturas de dados lineares em Python, com foco na modularidade, eficiência e reutilização. Foram implementadas pilhas, filas, arrays dinâmicos, listas encadeadas simples e duplas, listas circulares e filas de prioridade, além de aplicações práticas como Torre de Hanoi, calculadora de matrizes e simulações de filas.

1 Introdução

Este trabalho teve como objetivo implementar uma hierarquia de estruturas de dados lineares, utilizando programação orientada a objetos em Python. A modularização do código e a reutilização das classes foram prioridades para facilitar manutenções e extensões futuras.

2 Estruturas de Dados Utilizadas

2.1 Classe Abstrata Base

Foi criada uma classe abstrata para definir a interface comum das estruturas lineares, garantindo uniformidade e facilitando a extensão das classes.

2.2 IndexedArray

Implementação de um array dinâmico com capacidade de expansão automática e suporte a indexação via operadores, facilitando o acesso e modificação dos elementos.

2.3 Lista Encadeada Simples e Dupla

Listas encadeadas foram implementadas para armazenar dados de forma dinâmica. A lista dupla inclui métodos adicionais como troca de elementos adjacentes e ordenação por bubble sort.

2.4 Pilha e Fila

Pilhas e filas foram implementadas utilizando tanto arrays dinâmicos quanto listas encadeadas, permitindo comparar desempenho e capacidade de expansão.

2.5 Lista Circular Simples

Implementada para armazenar sequências circulares de dados, aplicada no armazenamento de figuras geométricas planas com cálculo de perímetro.

2.6 Fila de Prioridade

Baseada em lista dupla ordenada, mantém os elementos ordenados por prioridade, garantindo remoção eficiente do maior elemento.

3 Divisão de Módulos

O projeto foi organizado em um módulo principal contendo todas as classes de estruturas de dados (*linear_structures.py*) e módulos separados para aplicações práticas :

`torre_hanoi.py`: implementação da Torre de Hanoi usando pilha encadeada.

`calculadora_matrizes.py`: operações matriciais usando IndexedArray.

`fila_bandeja.py`: simulação de fila com controle de tempo.

`fila_prioridade.py`: fila de prioridade com lista dupla ordenada.

`simple_circular_list.py`: lista circular para armazenamento de pontos geométricos.

4 Descrição das Rotinas e Funções

Cada estrutura possui métodos para inserção, remoção, atualização e consulta, com tratamento de exceções para casos inválidos. Métodos especiais foram implementados para sobrecarga de operadores e ordenação.

5 Complexidades de Tempo e Espaço

- Arrays dinâmicos: acesso em tempo constante, inserção e remoção em tempo linear devido à movimentação de elementos.
- Listas encadeadas: inserção e remoção em tempo constante na cabeça, acesso linear.
- Ordenação por bubble sort: complexidade quadrática, adequada para listas pequenas.
- Pilhas e filas: operações em tempo constante.

6 Problemas e Observações

A manutenção correta dos ponteiros nas listas encadeadas foi um desafio, especialmente na lista dupla. A atualização dinâmica dos tempos na fila bandeirão exigiu cuidado para garantir consistência.

7 Conclusão

O projeto consolidou conhecimentos fundamentais em estruturas de dados e programação orientada a objetos, além de demonstrar a importância da modularização para facilitar a manutenção e expansão do código.