



— Aula de Estrutura de dados I — Unidade 2

Exercício 1: Análise de Complexidade de Algoritmos

Implemente dois algoritmos diferentes para encontrar o maior elemento em um vetor de inteiros. O primeiro algoritmo deve ter complexidade $O(n)$ e o segundo $O(n \log n)$. Após implementar ambos, escreva um programa em C que compare o tempo de execução dos dois algoritmos para vetores de diferentes tamanhos (por exemplo, 10^3 , 10^4 , 10^5). Utilize a biblioteca `ctime.h` para medir o tempo de execução.

Dicas:

- Para o algoritmo $O(n \log n)$, utilize uma abordagem de divisão e conquista (como o algoritmo de mergesort, por exemplo), e depois acesse o maior elemento).

Exercício 2: Implementação de um Tipo Abstrato de Dado (TAD) - Funções com Alocação Dinâmica

Implemente uma estrutura de dados do tipo Funções em uma linguagem de programação, utilizando um Tipo Abstrato de Dado (TAD). A TAD deve conter as seguintes operações:

- Adicionar Função:** Adicionar um novo função ao vetor. O função deve conter os seguintes campos:
 - Nome (string)
 - ID (inteiro)
 - Salário (float)
- Remover Função:** Remover um função da lista com base no ID.
- Buscar Função:** Buscar e retornar os dados de um função com base no ID.
- Listar Funções:** Exibir todos os funções presentes na lista, ordenados por ID ou Nome.
- Libertar Memória:** Libertar toda a memória alocada dinamicamente pelo vetor de funções ao final do programa.

1. Implementação das Funções:

- adicionarFunção:** Recebe os dados do função (nome, ID, salário) e adiciona ao vetor.
- removerFunção:** Remove o função com o ID especificado.
- buscarFunção:** Retorna os dados do função correspondente ao ID. Utilize um dos algoritmos apresentados em sala.
- listarFunções:** Exibe todos os funções, permitindo escolher a ordenação por ID ou Nome.
- libertarMemoria:** Liberta toda a memória alocada dinamicamente.

2. Programa Principal:

- Escreva um programa que utilize a TAD para manipular o vetor de funções.
- Permita que o usuário adicione, remova, busque e liste funções através de um menu interativo.

Exercício 3: Manipulação de Arquivos e Estruturas de Dados

Escreva um programa que leia um arquivo texto contendo dados de estudantes no seguinte formato: nome, idade, nota1, nota2, nota3

O programa deve:

- Armazenar os dados lidos em um vetor.
- Calcular a média de notas para o da estudante.
- Exibir na tela o nome e a média de cada estudante, organizando em ordem decrescente de média.
- Salvar os resultados em um novo arquivo chamado **resultados.txt**, mantendo a mesma ordem de exibição.

Dicas:

- Utilize a função `fopen` para abrir e criar arquivos, e `fscanf` ou `fgets` para ler os dados.
- Implemente uma função de ordenação (como o *bubble sort*) para organizar a lista de acordo com a média.

Arquivo de entrada:

```
Alice,20,8.5,7.0,9.2
Bob,22,6.5,8.0,7.5
Carol,19,9.0,9.5,10.0
David,21,7.0,6.5,8.0
Eve,20,9.5,8.8,9.0
Frank,23,5.0,6.2,7.5
Grace,22,8.8,7.7,8.4
Hannah,21,9.1,9.0,9.3
Irene,19,7.5,6.5,7.0
Jack,20,8.0,7.2,7.8
Kathy,23,6.5,7.0,6.8
Leo,22,9.2,8.5,9.0
Mia,21,7.8,8.2,7.9
Nathan,20,8.3,8.1,8.7
Olivia,19,9.4,9.6,9.5
Paul,23,6.9,7.4,7.2
Quinn,22,8.7,8.4,8.5
Rita,21,7.0,7.2,7.1
Sam,20,9.0,8.7,9.1
Tina,19,8.2,8.0,8.4
```