

Universidade Federal Fluminense
Instituto de Computação
Curso de Sistemas de Informação
Estruturas de Dados para SI

Primeira Lista de Exercícios

Análise de Complexidade

- 1) Disponha as seguintes funções em ordem crescente de complexidade assintótica: $\log n$, $\log \log n$, n^2 , $n \log n$, $n!$, n^n , 2^n , n , \sqrt{n} ,
- 2) Exiba exemplos de funções que sejam $O(n)$, $O(n \log n)$, $O(n^3)$, $O(2^n)$.

Recursividade

- 1) Implemente em C um algoritmo recursivo para o problema das Torres de Hanói. Analise a complexidade do algoritmo implementado.
- 2) Implemente um algoritmo baseado em Backtracking que compute todas as sequências de n bits. Analise a complexidade do algoritmo implementado.
- 3) Faça uma pesquisa sobre o algoritmo Quicksort. Implemente em C o algoritmo QuickSort. Mostre que a complexidade de pior caso para o Quicksort é $O(n^2)$.

Listas Lineares Sequenciais

- 1) Implemente um modulo em C para manipulação de Listas Lineares Sequenciais criadas por alocação dinâmica. Considere o tipo nó abaixo:

```
typedef struct _no{
    int chave;
    char * info;
}no_t;
```

- 2) Implemente a operação de busca de um nó com valor de chave igual a **valor**.
- 3) Implemente a operação ordenar. Adapte a solução do exercicio 3 em Recursividade.
- 4) Implemente a operação de busca binária.

- 5) Implemente as operações de inserção de um nó.
- 6) Implemente as operações de remoção de um nó com valor de chave igual a **valor**.
- 7) Implemente a operação de merge de duas listas lineares sequenciais ordenadas.
- 8) Implemente a operação de inversão.

Pilhas e Filas com alocação Sequencial

- 7) Implemente um módulo C para armazenamento e manipulação de pilhas usando alocação sequencial.
- 8) Implemente um módulo C para armazenamento e manipulação de filas usando alocação sequencial.

Listas Lineares Simplesmente Encadeadas.

- 1) Implemente um módulo em C para manipulação de Listas Lineares Simplesmente Encadeadas considerando o tipo nó abaixo:

```
typedef struct _no_smp{
    int chave;
    char * info;
    struct _no * prox;
}no_smp_t;
```

- 1) Implemente a operação de busca que retorne o ponteiro para um nó com valor de chave igual a **valor**.
- 2) Implemente as operações de inserção de um nó do tipo no_smp_t para os seguintes casos:
 - a) inserção no início.
 - b) inserção no final da lista.
 - c) inserção em ordem.
- 3) Implemente a operação de remoção de um nó com chave igual a **valor**.
- 5) Implemente uma operação que retorne um ponteiro para o n-ésimo nó de uma lista simplesmente encadeada.
- 6) Implemente uma operação que retorne um ponteiro para o n-ésimo nó de uma lista a partir do último.

Listas Lineares Duplamente Encadeadas

1) Implemente um módulo em C para manipulação de Listas Lineares Simplesmente Encadeadas considerando o tipo nó abaixo:

```
typedef struct _no_dup{
    int chave;
    char * info;
    struct _no_dup * ant;
    struct _no_dup * prox;
}no_dup_t;
```

1) Implemente a operação de busca que retorne um ponteiro para um nó da lista com valor de chave igual a valor.

2) Implemente a operação de inserção de um nó do tipo `no_dup_t` após um nó apontado por `ptr_no` na lista.

3) Implemente a operação de remoção de um nó apontado por `ptr_no`.

Aplicações

1) Considere uma expressão matemática contendo parênteses, colchetes e chaves. Verifique o balanceamento dos símbolos de agrupamento, isto é, se para cada símbolo que inicia uma sub-expressão existe o seu correspondente que a encerra.

2) Considere a uma expressão em notação polonesa reversa, por exemplo, $5\ 3\ +\ 2\ * \ .$. Escreva um a função que receba tal expressão e retorne o resultado de sua avaliação.

3) Implemente um programa no qual o usuário insere alguns caracteres em uma string com a possibilidade de efetuar comandos de fazer (**do**) e desfazer(**undo**).

4) Implemente uma função ordene um seqüência de inteiros usando duas pilhas.

5) Mostre como simular uma pilha usando somente uma fila.

6) Implemente duas pilhas usando um único *array*.

7) Escreva um programa que receba um arquivo texto e crie um índice em ordem alfabética das palavras do texto, indicando em que linhas elas ocorrem. Ignore pontuações.

8) Considere uma lista linear por encadeamento simples. Implemente uma função que verifica se a lista forma ou não um loop, isto é, se existe um nó que alcança algum já visitado na seqüência.

9) Considere uma lista encadeada simples. Usando somente uma varredura, encontre o nó na posição $\lfloor n/2 \rfloor$ onde n é o número de elementos da lista. (Dica:use dois ponteiros).

10) Suponha que você deve implementar um aplicativo de celular que exiba uma seqüência de fotos de uma coleção de forma circular e em ambos os sentidos. Identifique uma estrutura de dados que permita resolver os eu problema e justifique sua escolha.