

Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)
Estrutura de Dados I
Professor: Rodrigo Minetto
Lista de exercícios (seleção para revisão)

Exercício 1) Marque verdadeiro ou falso (cada resposta errada anula uma certa):

- () $n^2 = \mathcal{O}(n^3)$
- () $5 \times 2^n + n^6 - 10^{10} = \Theta(2^n)$
- () $2n + 1 = \mathcal{O}(n^2)$
- () $n \log n^3 = \Omega(2^n \log n)$
- () $n^k = \mathcal{O}(c^n)$, com $k > 1$ e $c > 1$ constantes.
- () $4^{\log n} = \Theta(n^2)$
- () $\log n^2 = \mathcal{O}(\log n)$ ($\log_b(x^n) = n \log_b x$.)
- () Se $f(n) = \mathcal{O}(g(n))$ e $g(n) = \mathcal{O}(h(n))$ então $f(n) = \mathcal{O}(h(n))$
- () Seja $f(n) = n^3 + n \lg n$ e $h(n) = n \lg n$. Então, $f(n) \in \Theta(h(n))$
- () $2^{n+1} = \mathcal{O}(2^n)$

Exercício 2) Escreva uma função iterativa e uma função recursiva que recebe uma lista **simplesmente** encadeada como entrada e retorna o **maior** elemento armazenado. Por exemplo, se $\ell = \{5, 6, 7, 0, 9, -2, 3, 2, 1, 0\}$ então o retorno da função é 9. A função deve utilizar o seguinte protótipo:

```
typedef struct node {
    int data;
    struct node *next;
} List;

int max_itr (List *l);

int max_rec (List *l);
```

Dica: você pode usar uma função externa para comparar dois números, desde que ela seja também apresentada.

Exercício 3) Escreva uma função recursiva que recebe uma lista **simplesmente encadeada** e um inteiro k como entrada e remove todas as ocorrências de k na lista. Por exemplo, se $\ell = \{1, 0, 1, 2, 1\}$ e $k = 1$, então após a remoção $\ell = \{0, 2\}$. A função deve utilizar o seguinte protótipo:

```
typedef struct node {
    int data;
    struct node *next;
} List;
```

```
List* remove_all (List *l, int k);
```

Ps. se o elemento não existir então a lista deve permanecer inalterada. Teste os limites da lista para ter certeza que sua função funciona, por exemplo, remover a cabeça, meio e cauda da lista.

Dica: esse é um exercício mais elaborado e que está presente na lista da aula de recursão. Na prova será cobrado um outro exercício com mesmo nível e que também está na lista da aula de recursão (todos podem cair com exceção do Sudoku).

Exercício 4) Projete uma estrutura de dados **pilha** baseada em uma lista com encadeamento duplo — considere uma lista com acesso direto a apenas a **cabeça** da lista. Precisamente, codifique as seguintes funções de forma iterativa:

```
typedef struct node {
    int data;
    struct node *next;
    struct node *prev;
} Stack;
```

```
Stack* push (Stack *l, int elem);
Stack* pop (Stack *l);
int peek (Stack *l);
int empty (Stack *l);
```

Exercício 5) Mostre como implementar uma FILA através da utilização de UMA OU MAIS PILHAS. Ou seja, codifique as funções enqueue e dequeue com operações de alto nível de um TAD Pilha (push, pop, ...).

```
void enqueue (Pilha *p, int elem) {
    ---> Como codificar uma operação enqueue com
    1 ou 2 pilhas e operações de push, pop ou empty?
}
```

```

int dequeue (Pilha *p) {
    ---> Como codificar uma operação dequeue com
    1 ou 2 pilhas e operações de push, pop ou empty?
}

```

Exercício 6) Foi descoberta uma espécie alienígena de ácido ribonucleico (popularmente conhecido como RNA). Os cientistas, por falta de criatividade, batizaram a descoberta de ácido ribonucleico alienígena (RNAA). Similar ao RNA que conhecemos, o RNAA é uma fita composta de várias bases. As bases são B , C , F , S e podem ligar-se em pares. Os únicos pares possíveis são entre as bases B e S e as bases C e F . Enquanto está ativo, o RNAA dobra vários intervalos da fita sobre si mesma, realizando ligações entre suas bases. Os cientistas perceberam que:

- quando um intervalo da fita de RNAA se dobra, todas as bases neste intervalo se ligam com suas bases correspondentes;
- cada base pode se ligar a apenas uma outra base e elas podem estar em posições consecutivas.
- as dobras ocorrem de forma a maximizar o número de ligações feitas sobre fitas.

A figura abaixo ilustra algumas dobras (mas elas não são únicas, ou seja podem ter dobras diferentes com o mesmo número de ligações).



Escreva um programa em linguagem C para determinar quantas ligações serão realizadas entre as bases se a tira de RNAA ficar ativa. É obrigatório usar um TAD, que faça diferença e seja útil, na solução do exercício. Você não precisa implementar as funções do TAD, basta usar e explicar o motivo.