

Universidade Federal de Itajubá – UNIFEI

Mestrado em Ciência e Tecnologia da Computação

*As questões a seguir são exemplos de problemas
que fizeram parte de provas de processos seletivos anteriores*

1. Em estruturas de dados como Grafos e Árvores, para a resolução de diversos problemas, é necessário percorrer os nós considerando algum critério de caminhamento. Uma forma de percorrer estes dados é através de uma *busca em profundidade*. No entanto, para se manter uma ordem de visita adequada, faz-se necessário utilizar outra estrutura de dados para armazenar os elementos que deverão ser visitados. A estrutura, neste caso, é a PILHA, ou seja, uma estrutura com a política de acesso LIFO (*Last-In-First-Out*), *último que entra, primeiro que sai*.

Pede-se descrever as seguintes funções para manipulação de uma PILHA:

- a) Função para inserir um elemento na Pilha.
- b) Função para retirar um elemento da Pilha.
- c) Função para verificar se a Pilha está vazia.

Observação: esta questão deve ser desenvolvida utilizando uma das seguintes linguagens de programação: C, C++ ou Pascal. Além disso, os elementos devem ser armazenados em um vetor, ou seja, a Pilha terá um limite de elementos que poderão ser armazenados.

2. Em estruturas de dados como Grafos e Árvores, para a resolução de diversos problemas, é necessário percorrer os nós considerando algum critério de caminhamento. Uma forma de percorrer estes dados é através de uma *busca em largura*. No entanto, para se manter uma ordem de visita adequada, faz-se necessário utilizar outra estrutura de dados para armazenar os elementos que deverão ser visitados. A estrutura, neste caso, é a FILA, ou seja, uma estrutura com a política de acesso FIFO (*First-In-First-Out*), *primeiro que entra, primeiro que sai*.

Pede-se descrever as seguintes funções para manipulação de uma FILA:

- a) Função para inserir um elemento na Fila.
- b) Função para retirar um elemento da Fila.
- c) Função para verificar se a Fila está vazia.

Observação: esta questão deve ser desenvolvida utilizando uma das seguintes linguagens de programação: C, C++ ou Pascal. Além disso, os elementos devem ser armazenados em um vetor, ou seja, a Fila terá um limite de elementos que poderão ser armazenados e, portanto, deverá ter tratamento circular.

3. Seja $N > 3$. Considere o trecho de algoritmo abaixo:

```

para i ← 1 até N-3 faça
|   w ← w + z + 10
|   para j ← 1 até N faça
|   |   z ← y + y + y + 30
|   fim-para
|   para k ← 1 até N-N faça
|   |   para l ← 1 até N faça
|   |   |   y ← x + 1
|   |   fim-para
|   |   para m ← 1 até N faça
|   |   |   x ← w + y + z
|   |   fim-para
|   fim-para
fim-para

```

Quantas operações "+" esse algoritmo realiza?

4. Considerando que foram inseridos, nesta ordem, os seguintes elementos em uma árvore binária de busca:

I, L, C, A, D, M, F, E, J, B, G, H

Pede-se a ordem de apresentação dos elementos, se a mesma árvore for percorrida em:

- a) Pré-ordem
- b) Pós-ordem

5. Considere a função em C++ para construção de uma árvore binária abaixo:

```

struct No {
    char Info;
    No *Esquerda;
    No *Direita;
};
void Constroi(No **A) {
    char D;
    cin >> D;
    if (D != '+') {
        *A = new No;
        (*A)->Info = D;
        Constroi (&(*A)->Esquerda);
        Constroi (&(*A)->Direita);
    } else *A = NULL;
}

```

Represente graficamente a árvore resultante se for inserida a sequência:

F E C + + D B + A + + + G + H I + + J + +

6. Desenvolva uma função (em C/C++ ou Pascal) chamada `InterseccaoArvBB` que recebe como parâmetro a raiz de duas Árvores Binárias de Busca e imprime no dispositivo de saída padrão as informações comuns que estão contidas em ambas as árvores. As árvores armazenam palavras de, no máximo, 50 caracteres.

Todas as funções e estruturas necessárias utilizadas devem ser apresentadas (implementadas).

7. Encontre a complexidade computacional para os seguintes laços:

```
a) for (int c1=0, i=1; i<=n; i++)
    for (int j=1; j<=n; j++)
        c1++;
```

```
b) for (int c2=0, i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        c2++;
```

```
c) for (int c3=0, i=1; i<=n; i*=2)
    for (int j=1; j<=n; j++)
        c3++;
```

```
d) for (int c4=0, i=1; i<=n; i*=2)
    for (int j=1; j<=i; j++)
        c4++;
```

8. Considerando a representação por listas de adjacências, escreva uma função (em C, C++ ou Pascal) que verifique se um grafo (passado como parâmetro) é regular. Apresente as declarações dos tipos usados.

9. Faça uma função que retorne o número de componentes conexas de um grafo simples representado através de sua matriz de adjacência. A função deverá receber como parâmetros a matriz e a sua respectiva ordem.

Observação: a função deve ser descrita em uma das seguintes linguagens de programação: C, C++ ou Pascal.

10. Considerando o grafo bipartido completo $K_{m,n}$, responda:

a) Para quais valores de m e n $K_{m,n}$ é euleriano?

b) Para quais valores de m e n $K_{m,n}$ é hamiltoniano?