



INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE ED e LAB. ED
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Avaliativa de ED e Lab. ED - Unidade III

Instruções

Responda às questões abaixo.

Atenção: **Escreva suas respostas na cor vermelha.**

Questões

Questão 1) Responda: (3 pontos)

a) O que é uma Fila e qual sua política de acesso aos dados?

R: Uma fila é uma estrutura de dados que enfileira os dados, possuindo os atributos de início e fim da fila para poder manipular os dados. Tendo como política de acesso aos dados, FIFO (First-in, First-out), ou seja, o primeiro elemento que for inserido dentro da estrutura, é o primeiro elemento a sair da estrutura.

b) O que é uma Pilha e qual sua política de acesso aos dados?

R: Uma pilha é uma estrutura de dados que empilha os dados, tendo como política de acesso aos dados, FILO (First-in, Last-out), ou seja, o primeiro elemento que for inserido dentro da estrutura, será o último elemento a sair da estrutura.

c) O que determina se uma Árvore Binária é uma Árvore Binária de Busca?

R: Se ela segue a regra das árvores binárias de busca, no caso, o nó inicial sendo a raiz, só pode ter ramificações a esquerda se forem menores que a raiz, e a direita se forem maiores que ela, ou seja, se temos o número 10 na raiz, e queremos inserir o número 8 e 11, o 8 seria uma ramificação da esquerda do 10 e o 11 ficaria a direita do mesmo, dessa forma seguiria a regra, valores menores ramificados a esquerda e valores maiores ramificados a direita.

Questão 2) Mostre como implementar uma Estrutura de Dados Fila usando duas Pilhas. (5 pontos)

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 5

// A IMPLEMENTAÇÃO DAS PILHAS

typedef struct sPilha
{
    int topo;

    int itens[MAX];
} Pilha;

void inicializarPilha(Pilha *pilha)
{
    pilha->topo = -1;
}

int estaVazia(Pilha *pilha)
{
    return pilha->topo == -1;
}
```

```
int estaCheia(Pilha *pilha)

{

    return pilha->topo == MAX - 1;

}


void empilhar(Pilha *pilha, int valor)

{

    pilha->topo++;

    pilha->itens[(pilha->topo)] = valor;

}


int desempilhar(Pilha *pilha)

{

    return pilha->itens[(pilha->topo)--];

}


// A FILA AGORA AMEM

typedef struct sFilaComDuasPilhas

{

    Pilha pilha1;

    Pilha pilha2;

} FilaComDuasPilhas;
```

```
void inicializarFila(FilaComDuasPilhas *fila)
{
    inicializarPilha(&(fila->pilha1));
    inicializarPilha(&(fila->pilha2));
}

void enqueue(FilaComDuasPilhas *fila, int valor)
{
    empilhar(&(fila->pilha1), valor);
}

int dequeue(FilaComDuasPilhas *fila)
{
    if (estaVazia(&(fila->pilha2)))
    {
        while (!estaVazia(&(fila->pilha1)))
        {
            empilhar(&(fila->pilha2), desempilhar(&(fila->pilha1)));
        }
    }
}
```

```
return desempilhar(&(fila->pilha2));  
  
}  
  
int main()  
{  
  
    FilaComDuasPilhas fila;  
  
    inicializarFila(&fila);  
  
    enqueue(&fila, 1);  
  
    enqueue(&fila, 2);  
  
    enqueue(&fila, 3);  
  
    printf("Elemento que foi com Deus: %d\n", dequeue(&fila));  
  
    printf("Elemento que foi com Deus: %d\n", dequeue(&fila));  
  
    enqueue(&fila, 4);  
  
    printf("Elemento que foi com Deus: %d\n", dequeue(&fila));  
  
    printf("Elemento que foi com Deus: %d\n", dequeue(&fila));  
  
    return 0;  
}
```

Questão 3) (ENADE 2011): Suponha que se queira pesquisar a chave 287 em uma árvore binária de busca com chaves entre 1 e 1 000. Durante uma pesquisa como essa, uma sequência de chaves é examinada. Cada sequência abaixo é uma suposta sequência de chaves examinadas em uma busca da chave 287. (2 pontos)

I. 7, 342, 199, 201, 310, 258, 287

II. 110, 132, 133, 156, 289, 288, 287

III. 252, 266, 271, 294, 295, 289, 287

IV. 715, 112, 530, 249, 406, 234, 287

É válido apenas o que se apresenta em:

A) I.

B) III.

C) I e II. X

D) II e IV.

E) III e IV.

Explique sua resposta!

R:

ÁRVORE BINÁRIA 1

7
342
199
201
310
258
287

ÁRVORE BINÁRIA 2

110
132
133
156
289

288
287

ÁRVORE BINÁRIA 3

252
266
271
294
289 295
287

ÁRVORE BINÁRIA 4

715
112
530
249
234 406
287

Como é possível ver, nas árvores 1 e 2, o caminho para chegar até o 287 é feito de maneira linear, ou seja, sem ter ramos descontinuados, só foram utilizados nós necessários para chegar até o 287, seguindo a regra das árvores binárias, onde os valores maiores ficam a direita, enquanto os valores menores ficam à esquerda do nó respectivo, as árvores 1 e 2 seguem perfeitamente essas regras, enquanto na árvore 3 e 4, isso não se qualifica, pois existem ramos descritos na sequência de chaves que não são necessários para alcançar o 287, na árvore 3 a ligação do 294 para o 295 e na árvore 4 a ligação do 249 para o 406, apesar de estarem sendo seguidas as regras de uma árvore binária corretamente, essas ligações não são necessárias em uma busca binária para alcançar o 287 nas árvores, tornando essas duas árvores inválidas.