Faculdade de Tecnologia da Zona Leste

Curso de Graduação em Análise e Desenvolvimento de Sistemas

Avaliação de Estrutura de Dados - 3º semestre período Noturno

Nome Completo: _____

Matrícula: _____

1) Dada a estrutura abaixo da classe NO, realize a implementação de um método **recursivo** que remova e retorne um número inteiro do final da Lista Duplamente Encadeada. O método implementado deve verificar se há elementos na lista, caso não haja, o mesmo deve retornar -1. (3,0 pontos)

```
public class NO {
    public int dado;
    public NO prox;
    public NO anterior;

    public NO(int e){
        dado=e;
        prox=null;
        anterior=null;
    }
}
```

2) Dadas as afirmações, assinale a alternativa que contém, de cima para baixo, a sequência correta, assinalando V para verdadeiro e F para falso. (0,25 pontos)

( ) A disciplina de acesso da estrutura de dados Pilha determina que o último elemento inserido no conjunto deva ser o primeiro a ser removido.

( ) A implementação de lista utilizando alocação sequencial dos elementos, comparada à alocação encadeada, necessita de mais espaço de armazenamento por elemento do conjunto.

( ) A pesquisa sequencial é mais eficiente que a pesquisa binária para busca de elementos em listas ordenadas implementadas com alocação sequencial dos elementos.

( ) As estruturas de dados Pilha e Fila podem ser implementadas utilizando tanto abordagens baseadas na alocação sequencial quanto na alocação encadeada dos elementos.

( ) A inserção de um elemento no início de uma lista duplamente encadeada implica no deslocamento dos elementos já existentes na memória.

a) V, V, V, F, F.

b) V, F, V, F, F.

c) V, F, F, V, F.

d) F, V, F, V, V.

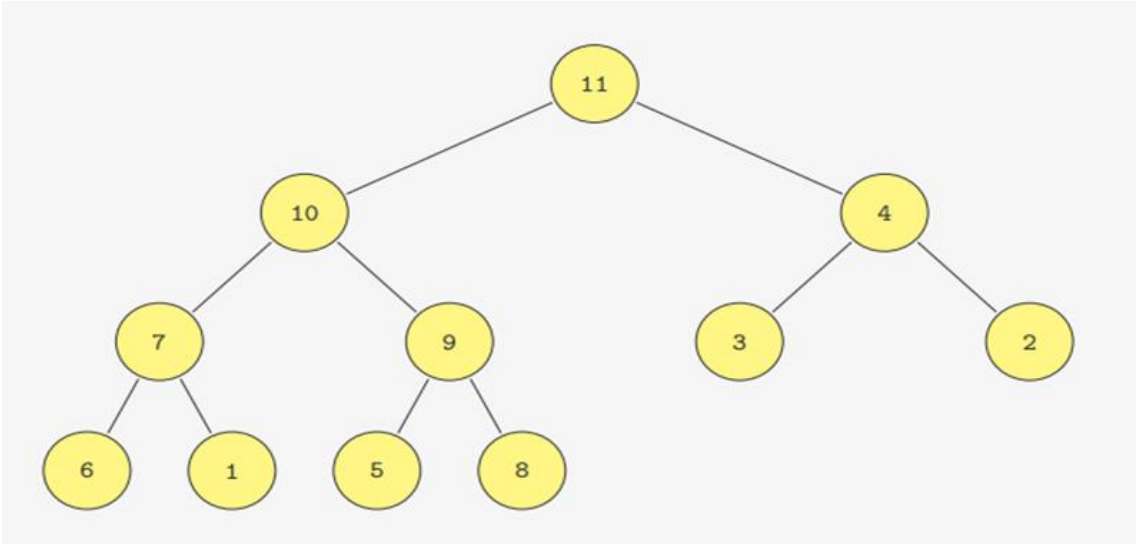
e) F, F, V, F, V.

3) Dado o algoritmo MergeSort, explique o funcionamento dos métodos abaixo e simule a sua execução para o seguinte domínio de entrada: [56, 29, 35, 42, 15, 41, 75, 21]  (2,5 pontos)

```
public static void mergeSortRecursivo(int lista[], int inicio, int fim){
    if (inicio < fim){
        int meio = (inicio + fim) / 2;
        mergeSortRecursivo(lista,inicio, meio);
        mergeSortRecursivo(lista,meio + 1, fim);
        mesclar(lista,inicio, meio, meio+1,fim);
    }
}

public static void mesclar(int lista[], int inicioA, int fimA,
        int inicioB, int fimB){
    int i1 = inicioA;
    int i2 = inicioB;
    int iaux = inicioA;
    int aux[] = new int[lista.length];
    while (i1 <= fimA && i2 <= fimB){
        if(lista[i1] <= lista[i2])
            aux[iaux++]=lista[i1++];
        else
            aux[iaux++]=lista[i2++];
    }
    while (i1 <=fimA)
        aux[iaux++]=lista[i1++];
    while (i2 <=fimB)
        aux[iaux++]=lista[i2++];
    for (int i=inicioA;i<=fimB;i++)
        lista[i] = aux[i];
}
```

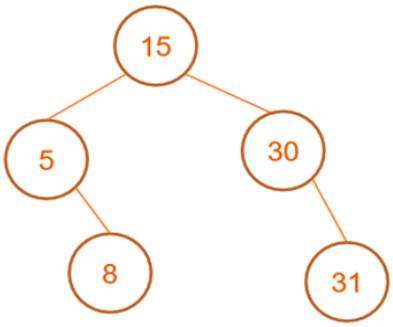
4) O algoritmo Heap Sort utiliza o conceito de Fila de Prioridades para realizar as operações de inclusão e remoção de elementos. Considerando a ordenação pelo Heap Máximo, **demonstre todos os passos** para a reordenação do algoritmo após a remoção de um elemento (1,5 pontos).




5) Com base no conceito de árvores binárias realize cada um dos exercícios abaixo (2,75 pontos):

a) Simule todos os passos até o estado final da árvore binária para os seguintes elementos: 25, 5, 30, 8, 20, 31, 3.

b) Dada a árvore binária abaixo, apresente o estado final da árvore ao realizar a remoção do Nó raiz.




c) Apresente os resultados das consultas dos Nós da árvore binária em pré-ordem e pós-ordem, respectivamente.