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 adicione um número inteiro no final da Lista Duplamente Encadeada. Se a lista estiver vazia a inserção deve ser realizada no início. Caso não esteja devem ser realizadas chamadas recursivas até que seja encontrado o último Nó da lista para a inclusão (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) Referente a estruturas de dados, é CORRETO afirmar: (0,25)

a) Uma lista encadeada é uma coleção linear de objetos de uma classe autoreferenciada, chamados de nós. Pode ser acessada por meio de um ponteiro para o primeiro nó da lista. Os nós subsequentes são acessados por meio do membro ponteiro de link armazenado em cada nó.

b) Por convenção, o ponteiro de link do último nó de uma lista é inicializado em 0 (zero).

c) O tamanho (quantidade de elementos) de uma lista encadeada deve ser definido na hora da criação.

d) Pilhas, filas e árvores são consideradas também estruturas de dados lineares, baseadas em listas encadeadas.

e) Uma pilha usa método de inserção FIFO

f) Todas as anteriores

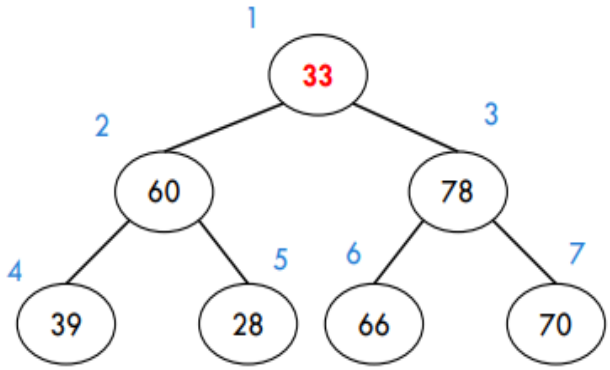
3) Dado o algoritmo MergeSort, explique o funcionamento dos métodos abaixo e simule a sua execução para o seguinte domínio de entrada: [53, 25, 32, 39, 16, 52, 98, 28] (2,5 pontos)

```
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).

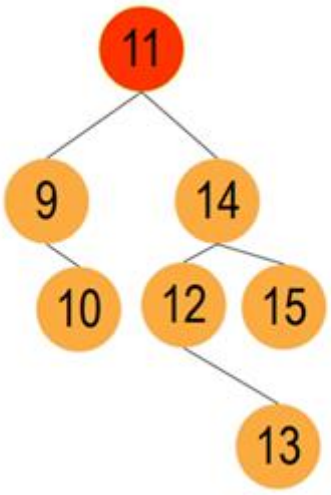
1	2	3	4	5	6	7
33	60	78	39	28	66	70



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: 15, 29, 5, 30, 7, 20, 98, 4.

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.