Prova de Estrutura de Dados
Professor Bernardo Henz

Nome:

1) A respeito de vetores em Java (ArrayList), assinale a alternativa **correta**:
a) Manipulação de elementos em vetores é lento, de forma que adicionar ou remover elementos implicam em fazer um shift de elementos no vetor
b) O acesso a um índice é lento, pois é necessário percorrer todos os elementos até o elemento desejado
c) Vetores são bastante usados em problemas que necessitam de alta manipulação de dados, visto que são otimizados para isso
d) Vetores não permitem duplicatas, pois utilizam uma função de hash

2) A respeito de listas, assinale a alternativa **correta**:
a) São estruturas sequenciais, de tamanho fixo, que são otimizadas para manipulação de dados
b) Para conseguir inserir/remover elementos no final da lista basta ter um ponteiro para o último Nodo
c) Conseguem inserir e remover elementos no início em tempo constante
d) O acesso ao elemento de um índice arbitrário é extremamente rápido, feito em tempo constante

3) Gabriela implementou uma Lista de inteiros otimizada que consegue efetuar as seguintes operações em tempo constante: adicionarNoInicio(int c), adicionarNoFinal(int c), removerDoInicio(int c). Utilizando esses métodos, implemente as classes Fila e Pilha:

```
public class Fila extends Lista {
    public void add(int c) {
        this. [1] (c);
    }
    public int remove( ) {
        return this. [2] ( );
    }
}
public class Pilha extends Lista {
    public void add( int c) {
        this. [3] (c);
    }
    public int pop( ) {
        return this. [4] ( );
    }
}
```

Assinale a alternativa que completa corretamente as lacunas [1], [2], [3] e [4]:
a) adicionarNoInicio, adicionarNoFinal, removerNoInicio, adicionarNoInicio
b) adicionarNoFinal, removerNoInicio, adicionarNoInicio, removerDoInicio
c) adicionarNoInicio, removerDoInicio, adicionarNoFinal, removerNoInicio
d) adicionarNoInicio, removerDoInicio, removerDoInicio, adicionarNoFinal

4) Gabriela descobriu que existem classes de Pilha e Fila em Java, e está praticando para aprender o funcionamento delas. Ajude ela dizendo qual saída do código abaixo:

```
public static void main (String[] args){
    Queue<Integer> minhaFila = new LinkedList<Integer>();
    Stack<Integer> minhaPilha = new Stack<Integer>();
    minhaFila.add(50);
```

```
        minhaFila.add(5);
        minhaPilha.add(30);
        minhaPilha.add(20);
        System.out.println(minhaPilha.pop());
        System.out.println(minhaFila.remove());
        minhaFila.add(19);
        minhaPilha.add(23);
        System.out.println(minhaPilha.pop());
        System.out.println(minhaPilha.pop());
        System.out.println(minhaFila.remove());
}
```

Qual é a sequência de inteiros que aparecem na saída do código?
a) 50, 30, 23, 20,19
b) 20, 50, 23, 30, 5
c) 20, 50, 30, 5, 23
d) 23, 50, 20, 30, 5

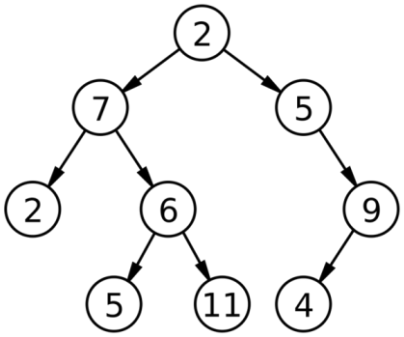
5) A respeito de Pilhas e Filas, assinale a alternativa **correta**:
a) As ações de inserir/remover são mais otimizadas em Filas, visto que Pilhas necessitam percorrer todos os elementos para retornar o elemento
b) As ações de inserir/remover são mais otimizadas em Pilhas, visto que Filas necessitam percorrer todos os elementos para retornar o elemento
c) As ações de inserir/remover são otimizadas igualmente em Pilhas e Filas, levando em média O(n/2) para executar essas operações (onde n representa o número de elementos da Pilha/Lista)
d) As ações de inserir/remover são otimizadas igualmente em Pilhas e Filas, levando O(1) para executar essas operações, ou seja, em tempo constante

6) Com respeito a listas duplamente encadeadas, assinale a alternativa **correta**:
a) Além do ponteiro para o próximo Nodo, também armazena ponteiro para o Nodo anterior, o que permite que o método 'removerDoFinal' possa ser realizada em tempo constante (quando armazenado um ponteiro para o último Nodo)
b) Devido à adição do ponteiro para o Nodo anterior, a lista dobrará o número de Nodos se comparado a uma Lista simples
c) Além de possuir ponteiro para o Nodo anterior, o último Nodo aponta a o primeiro
d) O tempo para percorrer uma Lista duplamente encadeada é o dobro de uma Lista simplesmente encadeada

7) A respeito de grafos, assinale a alternativa **correta**:
a) Todo grafo é uma árvore
b) Uma árvore binária é um grafo acíclico
c) Uma árvore é um grafo direcionado, e pode ou não ser cíclico
d) Grafos são estruturas lineares compostos por Nodos e igual número de Arestas

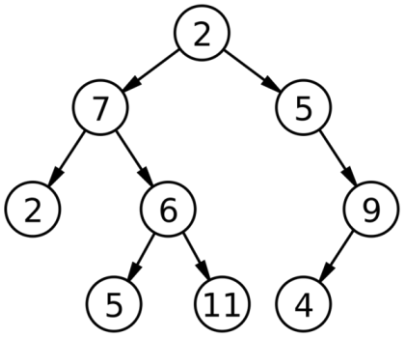
8) Com respeito a árvores, assinale a alternativa **correta**:
a) Todas as árvores devem estar ordenadas, ou seja, chaves menores para a esquerda, e chaves maiores para a direita
b) As folhas de uma árvore são todos os nodos que têm no mínimo um filho
c) Em uma árvore binária, cada Nodo pode ter no máximo 2 filhos
d) A profundidade de uma árvore é sempre a metade da quantidade de Nodos

9) Dada a seguinte árvore



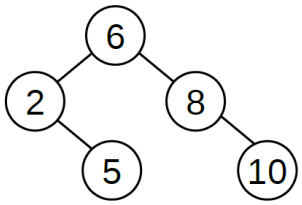
Assinale a alternativa **correta**:
a) Esta é uma árvore binária ordenada
b) Esta é uma árvore AVL
c) Esta é uma árvore balanceada
d) Esta árvore tem profundidade 3

10) Utilizando a mesma imagem anterior



Assinale a saída que imprime os elementos em **pré-ordem**:
a) 2, 7, 2, 6, 5, 11, 5, 9, 4
b) 2, 7, 5, 2, 6, 9, 5, 11, 4
c) 2, 7, 5, 6, 11, 2, 5, 9, 4
d) 7, 2, 5, 2, 6, 9, 5, 11, 4

11) Considere a seguinte Árvore AVL:



Considere que queiramos adicionar o valor 9 (lembre-se de verificar e rebalancear caso necessário). Qual será a saída caso a árvore seja impressa em **pós-ordem**:
a) 2, 5, 6, 8, 9, 10
b) 5, 2, 9, 10, 8, 6
c) 5, 2, 8, 10, 9, 6
d) 5, 2, 9, 10, 8, 6

12) A respeito de Java Collections, marque a alternativa correta:
a) Uma PriorityQueue, como o nome diz, segue a regra FIFO (first-in first-out), removendo sempre o elemento mais antigo
b) Um HashSet não permite duplicatas, diferente da LinkedList que permite
c) Apesar de Queue ser uma interface, conseguimos instanciar um objeto desse tipo
d) LinkedList herda tanto da classe List quanto da classe Queue