

EINF MA CID

Disciplina: Estruturas de Dados e Algoritmos I-2019/2020

Prova: mini-teste 3 (26-11-2019)

Esta prova tem a duração de **1 hora** e é **sem consulta**. Identifique **TODAS** as folhas de teste.

1. Considere que no código Java, que implementa as **ABP**, foi definido o método `q1`, como a seguir se apresenta:

<pre> public void q1{ q1(raiz); } public void q1(NoABP<E> x){ if (x!=null){ if (x.esq==null && x.dir==null) remove(x.elemento); q1(x.esq); q1(x.dir); } } </pre>	<p>Suponha <code>t</code> a ABP obtida por inserção dos nós 13;15;10;6;14;31;40;34;19.</p> <p>(a) Qual o resultado da execução de <code>t.q1()</code></p> <p>(b) Qual a complexidade do método <code>q1</code>?</p> <p>(c) Se quiser aplicar o método a uma árvore binária que alterações deverá fazer no código apresentado?</p>
---	---

2. Nas mesmas condições do exercício anterior foi definido o método `q2` :

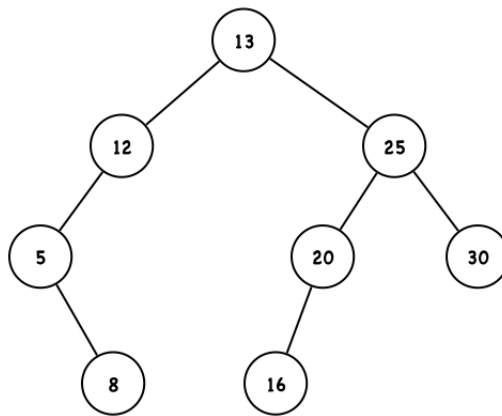
<pre> public LinkedList<E> q2(E x){ return q2(x, raiz); } public LinkedList<E> q2(E x, NoABP<E> n) { LinkedList<E> l1 = null; if (n == null) return new LinkedList<E>(); else { if (x.compareTo(n.elemento) < 0) { l1 = q2(x, n.esq); if (l1.isEmpty()){ return l1;} else l1.add(l1.size(), n.elemento); } else { if (x.compareTo(n.elemento) >0) { l1 = q2(x, n.dir); if (l1.isEmpty()) { return l1;} else l1.add(0, n.elemento); } else{ l1 = new LinkedList<E>(); l1.add(0, n.elemento);} } return l1; } } </pre>	<p>Seja <code>t</code> a árvore do exercício de 1</p> <p>(a) Qual o resultado de <code>t.q2(40)</code>?</p> <p>(b) Qual o resultado de <code>t.q2(15)</code>?</p> <p>(c) Qual a complexidade do método <code>q2</code>?</p>
--	---

3. Considere que na classe **ABP**, definiu o método **q3**

```
public void q3(){
    q3(raiz);
}
private void q3(NoABP<E> n) {
    System.out.print(" [");
    Queue<NoABP<E>> it=new QueueArray<>( );
    it.enqueue(n);
    while(!it.empty()){
        NoABP<E> x=it.dequeue();
        if (x.esq()!=null)
            it.enqueue(x.esq());
        if (x.dir!=null)
            it.enqueue(x.dir);
        System.out.print(x.elemento+" ");
    }
    System.out.println( "]" );
}
```

- (a) Qual o resultado de **t.q3()**, para **t** a árvore da questão 1?
- (b) Qual o resultado de **t.q3()**, para **t** a árvore da questão 4?
- (c) Qual a complexidade do método **q3()**

4. Considere a seguinte árvore:



- (a) Trata-se duma ABP?
- (b) Por que ordem deve inserir os nós para obter a árvore da figura?
- (c) Trata-se duma AVL? Justifique.
- (d) Qual a árvore obtida se remover o nó 5? E seguidamente remover o nó 8?
- (e) Sobre a árvore obtida na alínea anterior insira os nós pela ordem que se apresentam, identificando o nó em desequilíbrio, qual o caso da rotação, e qual a árvore resultante após as rotações: 29;27;5;1;7;35

5. Implemente para a classe **ABP** o método **void filhosUnicos()** que lista os filhos únicos da **ABP**

6. Implemente para a mesma classe o método **LinkedList<E> folhasCrescente()** que retorna uma lista com todas as folhas da árvore, mas ordenadas por ordem crescente. Por exemplo para a árvore da questão 4 o método deve retornar **[8,16,30]**