

Número:

Nome:

EINF MA CID

Disciplina: Estruturas de Dados e Algoritmos I-2016/2017

Prova: Teste 2 (17-11-2016)

Esta prova tem a duração de **2 horas** e é **sem consulta**. Identifique TODAS as folhas de teste.

1. Sejam L e P, duas listas que contêm inteiros ordenados por ordem crescente. A operação *printLots* irá listar os elementos de L, que estão nas posições especificadas em P. Por exemplo se P for a lista {1, 3, 4, 6}, o método *printLots(L,P)*, deve listar o 1º, 3º 4º e 6º elemento da lista L. Apresente uma implementação para o método *printLots*, usando unicamente as operações da API das Colecções.

```
public static void printLots(List<Integer> L, List<Integer> P){
```

```
}
```

2. Implemente o método *contains(Type x)* como método de instância, da classe *LinkedList<Type>*. O método deve retornar true, se a x pertence à lista, e false, caso contrário. A classe *LinkedList* é a classe que implementa listas ligadas usando "Single Nodes".

```
public boolean contains(Type x){
```

```
}
```

3. Um iterador para listas(objecto do tipo *ListIterator*) extende as funcionalidades de *Iterator* para as collecções. Os métodos *previous()* e *hasPrevious()* permitem a travessia das listas do fim para o início. Considere que a classe *MyDoubleListIterator<T>* implementa estes iteradores, usando nós duplamente ligados. Defina para esta classe os métodos, abaixo, considerando que estão definidas para a classe 3 variáveis de instância:

```
DoubleNode<T> current; // nó actual  
DoubleNode<T> beginMarker; // dummy node início  
DoubleNode<T> endMarker; // dummy node fim
```

(a)	public boolean hasPrevious(){ }
(b)	public T previous(){ }
(c)	public void remove(){ }

4. Insira numa ABP inicialmente vazia, os nós 14,10,97,4,32,7,5,34,11,50, pela ordem que se apresentam. Apresente a árvore obtida

5. Remova os nós, 10 depois o 14 e depois o 54 e apresente as árvores resultantes das remoções:

remove(10)	remove(14)	remove(54)

6. Apresente a listagem para a árvore final da alínea anterior em

(a) pré-ordem:

(b) pós-ordem:

7. Conceba um algoritmo que teste se todos os nós duma dada árvore binária satisfazem a propriedade imposta aos nós das árvores binárias de pesquisa.

8. Implemente o método de instância para ABP's *altura()*, que retorna a altura da árvore. Recorde que a altura duma árvore é a altura da sua raiz, e que a altura dum nó é o comprimento do caminho mais longo do nó até uma folha.