

MAC0121 - Algoritmos e Estruturas de Dados I
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO – SEG. SEMESTRE DE 2018
Terceira Prova – 29 de novembro de 2018

Nome do aluno: _____

Assinatura: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. Preencha o cabeçalho acima.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade.
4. A prova consta de 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido a consulta a livros, apontamentos ou colegas.
7. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 2 horas

Questão	Nota
1	
2	
3	
Total	

1. (valor 3.0 pontos)

Considere as seguintes afirmações sobre árvores de busca binária. Para cada afirmação abaixo, diga se é falsa ou verdadeira, **justificando detalhadamente**:

- (a) Ao procurar a chave 300 em uma árvore de busca binária foram visitados os seguintes nós na ordem abaixo:

512, 117, 410, 128, 225, 288, 312, 210, 300.

- (b) Se a árvore foi construída a partir de uma árvore vazia com os nós na seguinte ordem

52 37 18 21 15 61 52 19 ... 35 ... 20 ...

Na ABB construída com estes elementos, 21 é ancestral de 35, mas não é ancestral de 20.

- (c) A árvore terá altura máxima $n - 1$ em dois casos: quando os elementos foram incluídos em ordem crescente ou em ordem decrescente. Portanto, a probabilidade de que a altura da árvore seja $n - 1$ é

$$\frac{2}{n!}$$

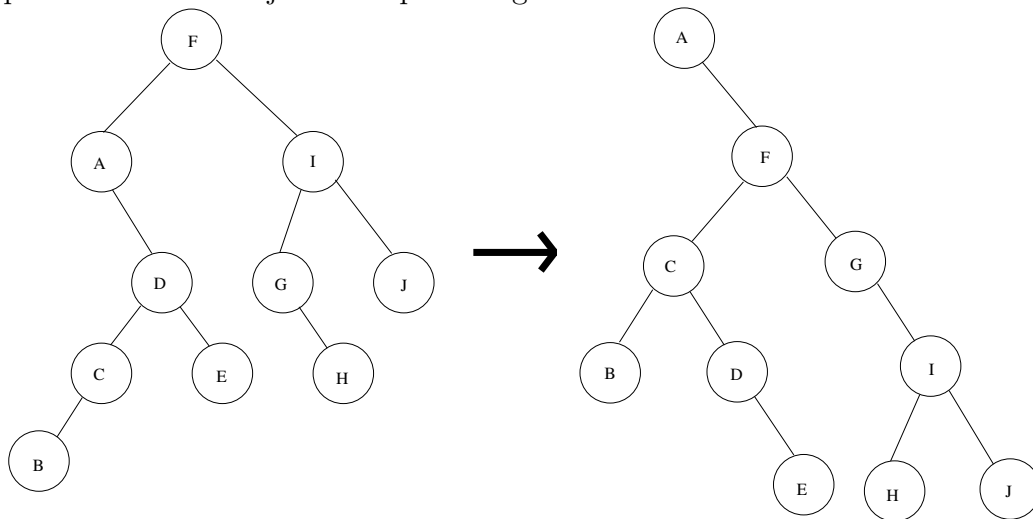
- (d) Considere o conjunto B de vértices do caminho visitado na árvore a partir da raiz quando uma folha é procurada. Tome agora o conjunto A de vértices à esquerda deste caminho e o conjunto C de vértices à direita do caminho. Então, para quaisquer $a \in A$, $b \in B$ e $c \in C$, temos $a < b < c$.

2. (valor 3.5 pontos)

Dizemos que uma árvore de busca binária **pende a direita** se para cada um dos seus nós, a altura da subárvore esquerda é menor ou igual do que a altura da subárvore direita. Faça uma função `pendeDireita` com o protótipo

```
apontador pendeDireita(apontador raiz)
```

que recebe uma árvore de busca binária e faz rotações à direita em seus nós de forma que ela penda a direita. Veja o exemplo da figura abaixo.



Você pode usar, sem escrever, funções para calcular a altura de uma árvore binária e para rodar uma árvore binária à direita, como vistas em aula, de protótipos abaixo:

```
int altura(apontador raiz);
apontador rodaDir (apontador raiz);
```

Qual a complexidade de sua função? Justifique.

3. (valor 3.5 pontos)

Faça uma função **recursiva** de protótipo

```
void listaIntervalo (apontador raiz, int inf, int sup)
```

que recebe uma árvore de busca binária apontada por **raiz** e dois números inteiros, e lista **em ordem crescente** todos os nós da árvore que estão no intervalo `[inf, sup]`. Qual a complexidade de sua função? Justifique.

