

- 1) Implemente um método, para uma **Árvore Binária de Busca**(cujos dados armazenados são **inteiros**), que retorne a soma de todos os elementos da árvore (2,0)

- 2) Implemente um método, para uma **Árvore Binária de Busca**, que retorne o nível em que se encontra determinado valor (passado como parâmetro). Considere o nível da raiz como sendo 1 (um). Caso o valor não esteja na árvore, retorne zero (2,5)

Questão 2 (2,5 pontos)

Utilizando os conhecimentos adquiridos até o momento, resolva o que se pede.

Implemente um TAD para um dicionário de sinônimos, partindo do seguinte princípio: cada palavra contida no dicionário possui no máximo 5 sinônimos diferentes.

Obs.:

Aluno: “Professor, quais métodos devem ser criados?”

Professor: “Aqueles que forem necessários para bem utilizar este TAD”

Questão 3 (3,4)

Escreva um método que imprima todos os nós de uma árvore binária que possuem **pelo menos** um filho.

Questão 1 (valor 1,5)

Responda aos seguintes itens, **SEMPRE JUSTIFICANDO SUAS RESPOSTAS**:

- a) Apresente o desenho de uma árvore binária de busca contendo os valores abaixo e que possua 5 (cinco) níveis (considere que a raiz está no nível 1) (0,5)

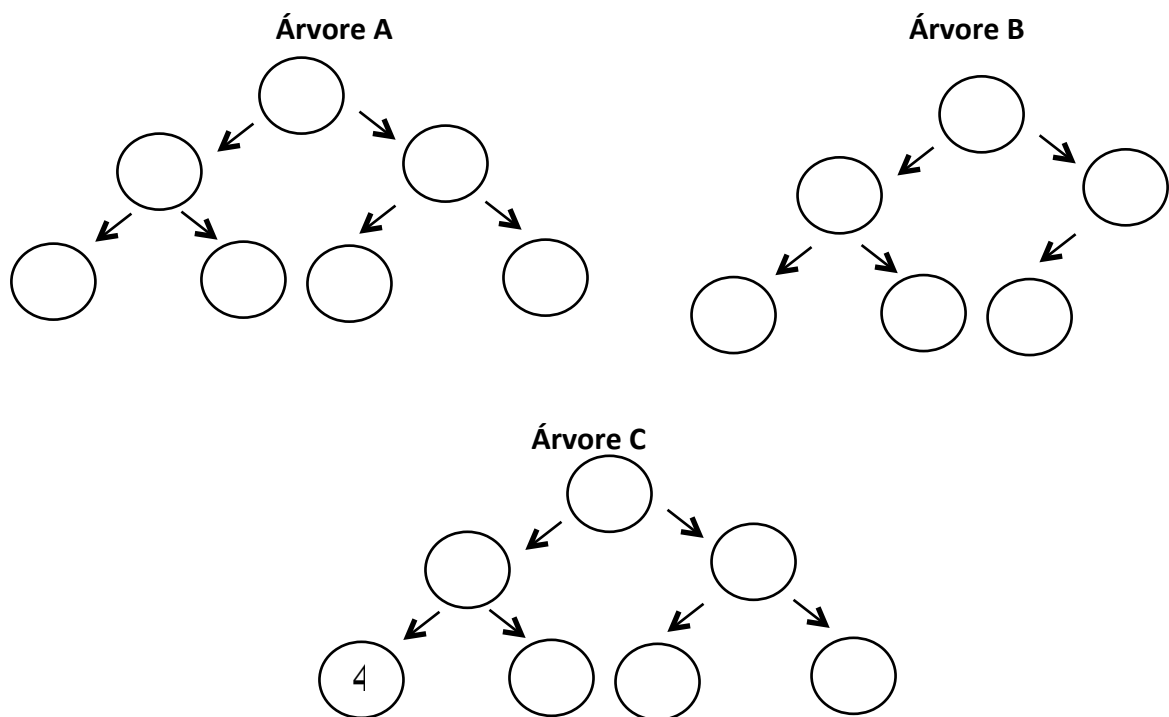
| | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Valores: | A | B | C | D | E | F | G | H |
| | I | J | K | L | M | N | P | Q |

- b) Considerando que uma árvore binária **completa** (todos os nós não folhas têm os dois filhos a que têm direito), com um nível possui um elemento; com dois níveis, possui três elementos; com três níveis, possui sete elementos; quantos elementos ela terá, com ***n*** níveis? Não se esqueça de justificar sua resposta, explicando como chegou a este resultado (0,5)

- c) Responda o mesmo do item b, considerando uma árvore **terciária** (cada nó tem no máximo três filhos) **completa** (0,5)

Questão 3 (3,0)

Duas árvores binárias são semelhantes em sua estrutura se ambas estiverem vazias, ou se forem não-vazias e suas subárvores esquerdas e subárvores direitas forem semelhantes. Implemente um método para determinar se duas árvores binárias são semelhantes, retornando true se forem semelhantes e false caso contrário. Atenção: a semelhança observada aqui é em relação à estrutura da árvore, **independente** do conteúdo de seus nós. A figura abaixo ilustra isso: Árvore A é semelhante à Árvore C, e não é semelhante à Árvore B



- 3) Implemente um método, para uma **Árvore Binária de Busca**, que retorne o menor elemento da árvore (2,0)

- 4) Implemente um método, para uma **Árvore Binária**, que retorne o maior elemento da árvore (2,0)
- 5) Implemente um método, para uma **Árvore Binária**, que imprima todos os valores maiores que determinado valor, passado como parâmetro (2,0)

Questão 2 (POSCOMP 2005) O número máximo de nós no nível i de uma árvore binária é: (Considere o nível da raiz igual a 1).

- a) 2^{i+1} , $i \geq 0$
- b) 2^{i-1} , $i \geq 1$
- c) 2^i , $i \geq 1$
- d) $2^i + 1$, $i \geq 1$
- e) $2^i - 1$, $i \geq 1$

Questão 3 (POSCOMP 2010) Os algoritmos a seguir representam os três caminhamentos para árvores binárias.

caminhamento(binário)

sebinário.esquerda \neq NULL então caminhamento(binário.esquerda)

escreverbinário.valor

sebinário.direita \neq NULL então caminhamento(binário.direita)

caminhamento(binário)

escreverbinário.valor

sebinário.esquerda \neq NULL então caminhamento(binário.esquerda)

sebinário.direita \neq NULL então caminhamento(binário.direita)

caminhamento(binário)

sebinário.esquerda \neq NULL então caminhamento(binário.esquerda)

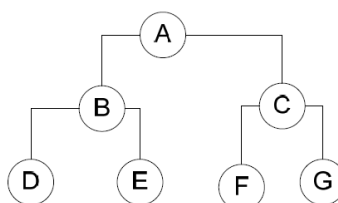
sebinário.direita \neq NULL então caminhamento(binário.direita)

escreverbinário.valor

Assinale a alternativa que contém os nomes dos 3 caminhamentos, respectivamente.

- a) pré-ordem, pós-ordem, em-ordem
- b) pré-ordem, em-ordem, pós-ordem
- c) pós-ordem, pré-ordem, em-ordem
- d) em-ordem, pré-ordem, pós-ordem
- e) em-ordem, pós-ordem, pré-ordem

Questão 5 (POSCOMP 2009) Percorrendo a árvore binária a seguir em pré-ordem, obtemos que sequência de caracteres?



- a) A C G F B E D
- b) G C F A E B D
- c) A B C D E F G
- d) D B E A F C G
- e) A B D E C F G

Questão 6 (POSCOMP 2004) Considere as seguintes definições de ordens de percurso de uma árvore binária:

Ordem A:

se a árvore binária não for vazia, então:

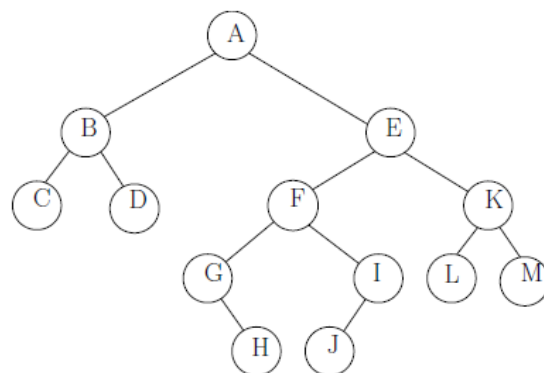
```
{visitar a raiz;
percorrer a sub-árvore esquerda em Ordem B;
percorrer a sub-árvore direita em Ordem B;
}
```

Ordem B:

se a árvore binária não for vazia, então:

```
{visitar a raiz;
percorrer a sub-árvore direita em Ordem A;
percorrer a sub-árvore esquerda em Ordem A;
}
```

Considere a seguinte árvore binária.



O percurso da árvore binária apresentada em **Ordem A** resulta em qual sequência de visitas?

- a) A B D C E K L M F I J G H
- b) A B C D E F G H I J K L M
- c) A B D C E K L M F G H I J
- d) A B E C D F K G I L M H J
- e) A B D C E F I J G H K L M

Questão 7 (POSCOMP 2005) Em uma estrutura de árvore binária de busca, foram inseridos os elementos h, a, b, c, i, j nesta sequência. O tamanho do caminho entre um nó qualquer da árvore e a raiz é dado pelo número de arestas (ligações entre os nós) neste caminho. Qual o tamanho do maior caminho na árvore, após a inserção dos dados acima?

- a) 2
- b) 6

- c) 4
- d) 5
- e) 3

Questão 9 Escreva um método que conte o número de nós em uma árvore binária que possuem **pelo menos** um filho.

Questão 2: (ENADE 2011) Suponha que se queira pesquisar a chave 287 em uma árvore binária de pesquisa com chaves entre 1 e 1000. Durante uma pesquisa como essa, uma sequência de chaves é examinada. Cada sequência abaixo é uma suposta sequência de chaves examinadas em uma busca da chave 287.

- I. 7, 342, 199, 201, 310, 258, 287
- II. 110, 132, 133, 156, 289, 288, 287
- III. 252, 266, 271, 294, 295, 289, 287
- IV. 715, 112, 530, 249, 406, 234, 287

É válido apenas o que se apresenta em

- A) I.
- B) III.
- C) I e II.
- D) II e IV.
- E) III e IV.

Questão 3: (ENADE 2008 - adaptado) Um programador propôs um algoritmo não-recursivo para o percurso em preordem de uma árvore binária com as seguintes características.

- Cada nó da árvore binária é representado por um registro com três campos: chave, que armazena seu identificador; esq dir, ponteiros para os filhos esquerdo e direito, respectivamente.
- O algoritmo deve ser invocado inicialmente tomando o ponteiro para o nó raiz da árvore binária como argumento.
- O algoritmo utiliza push() e pop() como funções auxiliares de empilhamento e desempilhamento de ponteiros para nós de árvore binária, respectivamente.

A seguir, está apresentado o algoritmo proposto, em que λ representa o ponteiro nulo.

```

Procedimento preordem (ptr:PtrNoArvBin)
  Var ptr :PtrNoArvBin;
  ptr := ptr;

  Enquanto (ptr  $\neq$   $\lambda$ ) Faça
    escreva (ptr.chave);
    Se (ptr.dir  $\neq$   $\lambda$ ) Então
      push(ptr.dir);
    Se (ptr.esq  $\neq$   $\lambda$ ) Então
      push(ptr.esq);

```

```

ptr := pop( );
Fim_Enquanto
Fim_Procedimento

```

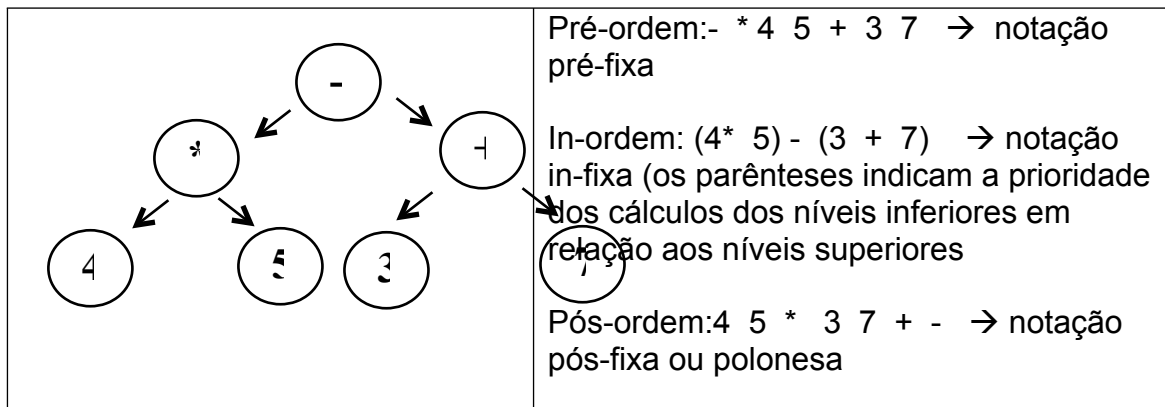
Com base nessas informações e supondo que a raiz de uma árvore binária com n nós seja passada ao procedimento `preordem()`, julgue os itens seguintes.

- I. O algoritmo visita cada nó da árvore binária exatamente uma vez ao longo do percurso.
- II. O algoritmo só funcionará corretamente se o procedimento `pop()` for projetado de forma a retornar `caso` a pilha esteja vazia.
- III. Empilhar e desempilhar ponteiros para nós da árvore são operações que podem ser implementadas com custo constante.
- IV. O último elemento a ser impresso é o descendente mais à direita da árvore.

Assinale a opção correta.

- A) Apenas um item está certo.
- B) Apenas os itens I e IV estão certos.
- C) Apenas os itens I, II e III estão certos.
- D) Apenas os itens II, III e IV estão certos.
- E) Todos os itens estão certos.

Questão 4: Considere uma “árvore de expressões”, como uma árvore binária como a apresentada abaixo. Os operandos estão nos nós-folha e os operadores aparecem nos nós-internos. Os percursos realizam o cálculo, como exemplificado a seguir:



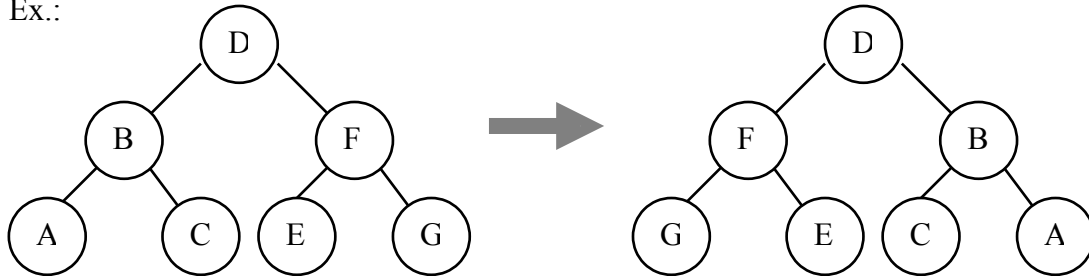
Implemente um método que retorne o cálculo de qualquer expressão apresentada no formato de uma árvore de expressões, de qualquer tamanho.

Obs.: Toda árvore de expressão estará corretamente estruturada, ou seja, com todos os operandos e operadores que compõem a expressão.

1. Escreva uma rotina que aceite um ponteiro para uma árvore binária e gere a sua imagem espelhada, conforme apresenta a ilustração abaixo.

Você deve fazer duas funções que realizem esta tarefa. A primeira deverá retornar a imagem espelhada em uma nova árvore, e a segunda deverá modificar a mesma árvore e retorná-la como sua própria imagem espelhada. (3,0)

Ex.:



. Sobre árvores balanceadas:

- a) Simule o algoritmo de inserção em uma **árvore binária de busca balanceada** para as seqüências abaixo mostrando, para **cada novo item inserido**, o estado da árvore (0,75 para cada seqüência):

Primeira seqüência: 29 1324 18 7 1 4 5 19 23

Segunda seqüência: 12 3 4 5 6 7 8 9 10 11 12

Caso necessite:

```
rotacao_esquerda (tree p);
```

```
q = p->dir;  
temp = q->esq;;  
q->esq = p;  
p->dir = temp;  
p = q;
```

```
rotacao_direita (tree p);
```

```
q = p->esq;  
temp = q->dir;;  
q->dir = p;  
p->esq = temp;  
p = q;
```

1. As seqüências abaixo são obtidas ao percorrermos uma **árvore binária** em pós-ordem e in-ordem (1,0).

pós-ordem: C I F B K L J H G E D A

in-ordem: A C B I F D G K J L H E

Construa (desenhe) a árvore binária correspondente.

Obs.: caso não seja possível construir uma árvore com esses percursos, justifique.

2. Em uma aplicação queremos utilizar uma **árvore binária de busca** para armazenar números inteiros e fazer as seguintes operações:

- a) Dados dois inteiros **a** e **b**, eliminar todos os nós **n** da árvore tais que $a \leq n \leq b$. (2,0)
- b) Dados dois inteiros **a** e **b**, verificar se eles estão na árvore e, caso estejam, retornar o primeiro ancestral comum a eles. (2,0)

Implemente as funções para realizar tais operações.

3. Sobre árvores balanceadas:

- b) Dê exemplos de **árvores binárias de busca** que são e que não são balanceadas. (0,5)
- c) Simule o algoritmo de inserção em uma **árvore binária de busca balanceada** para a seqüência abaixo mostrando, para **cada novo item inserido**, o estado da árvore (0,5):

1324 18 7 1 4 5 19 23 29

```
rotacao_esquerda (tree p);
```

```
q = p->dir;  
temp = q->esq;;  
q->esq = p;  
p->dir = temp;  
p = q;
```

```
rotacao_direita (tree p);
```

```
q = p->esq;  
temp = q->dir;;  
q->dir = p;  
p->esq = temp;  
p = q;
```


1. Duas árvores binárias são semelhantes se ambas estiverem vazias, ou se forem não-vazias e suas subárvores esquerdas e subárvores direitas forem semelhantes. Implemente uma função para determinar se duas árvores binárias são semelhantes (2,5).

Questão 1 (valor 4,0)

Implemente, utilizando recursividade, um método para uma árvore binária de busca que, recebendo como parâmetro um nó (este nó seria a raiz de uma árvore), apresente todos os elementos desta árvore nível a nível. Ou seja, apresenta a raiz, depois seus filhos, depois seus netos, e assim por diante. Obs. Não há necessidade de especificar quem é de que nível, podem ser apresentados um após o outro.

Questão 1 (valor 1,0)

Implemente um método para uma árvore binária de busca que, recebendo como parâmetro um nó (este nó seria a raiz de uma árvore), apresente todos os elementos desta árvore em ordem decrescente. Esta implementação deve utilizar recursividade.

Questão 2 (valor 1,0)

Considere uma árvore binária de busca que contém números inteiros. Implemente um método para esta árvore que, recebendo como parâmetro um nó (este nó seria a raiz da árvore), retorne a quantidade de números pares.

Questão 3 (valor 1,0)

Considere uma árvore binária de busca que contém números inteiros. Implemente, **de forma otimizada**, um método para esta árvore que, recebendo como parâmetros um nó (este nó seria a raiz da árvore) e um determinado valor, imprima todos os valores da árvore que são maiores que este.

Questão 4

Responda aos seguintes itens, **SEMPRE JUSTIFICANDO SUAS RESPOSTAS**:

- a) Apresente o desenho de uma árvore binária de busca contendo os valores abaixo e que possua 5 (cinco) níveis (considere que a raiz está no nível 1) (0,25)
Valores: **A** **B** **C** **D** **E** **F** **G** **H**
 I **J** **K** **L** **M** **N** **P** **Q**
- b) Considerando que uma árvore binária **completa** (todos os nós não folhas têm os dois filhos a que têm direito), com um nível possui um elemento; com dois níveis, possui três elementos; com três níveis, possui sete elementos; quantos elementos ela terá, com n níveis? Não esqueça de justificar sua resposta, explicando como chegou a este resultado (0,25)
- c) Responda o mesmo do item b, considerando uma árvore terciária (cada nó tem no máximo três filhos) completa (0,5)
- d) O método de fibonacci (11 2 3 5 8 13...) pode ser implementado de forma recursiva. Seus valores, a cada passo da execução, podem ser visualizados no

formato de uma árvore binária. Apresente esta árvore (0,25). Descreva a diferença entre um método ser recursivo e uma estrutura de dados (como a árvore, por exemplo) ser recursiva (0,75).