

Resolução Lista 9

1- Duas ABBs são SIMILARES se possuem a mesma distribuição de nós (independente dos valores nos mesmos). Em uma definição mais formal, duas ABBs são SIMILARES se são ambas vazias, ou se suas subárvores esquerdas são similares, e suas subárvores direitas também são similares. Implemente a função que verifica se duas árvores são similares.

Resposta:

```
int similares(Arvore *a1,Arvore *a2){
    return(similares_rec(a1->raiz,a2->raiz));
}

int similares_rec(No *p,No *q){
    if((p==NULL)&&(q==NULL))
        return 1;
    if((p==NULL) || (q==NULL))
        return 0;
    return((similares_rec(p->esq,q->esq))&&(similares_rec(p->dir,q->dir)));
}
```

2- Duas ABBs são IGUAIS se são ambas vazias ou então se armazenam valores iguais em suas raízes, suas subárvores esquerdas são iguais, e suas subárvores direitas são iguais. Implemente a função que verifica se duas árvores são similares.

Resposta:

```
int igualdade(Arvore *a1,Arvore *a2){
    return(igualdade_rec(a1->raiz,a2->raiz));
}

int igualdade_rec(No *p,No *q){
    if((p==NULL)&&(q==NULL))
        return 1;
    if((p==NULL) || (q==NULL))
        return 0;
    if(p->info.chave!=q->info.chave)
        return 0;
    return((igualdade_rec(p->esq,q->esq))&&(igualdade_rec(p->dir,q->dir)));
}
```

3- Uma ABB é estritamente binária se todos os nós da árvore tem 2 filhos. Implemente uma função que verifica se uma ABB é estritamente binária.

Resposta:

```
int estritamenteBin(Arvore *a){
    return(estritamenteBin_rec(a->raiz));
}

int estritamenteBin_rec(No *p){
    if(p==NULL)
        return 1;
    if((p->esq==NULL)&&(p->dir==NULL))
        return 1;
    if((p->esq==NULL) || (p->dir==NULL))
        return 0;
    return((estritamenteBin_rec(p->esq))&&( estritamenteBin_rec(p->dir))));
}
```

4- Implemente uma função para testar se uma árvore binária é uma ABB.

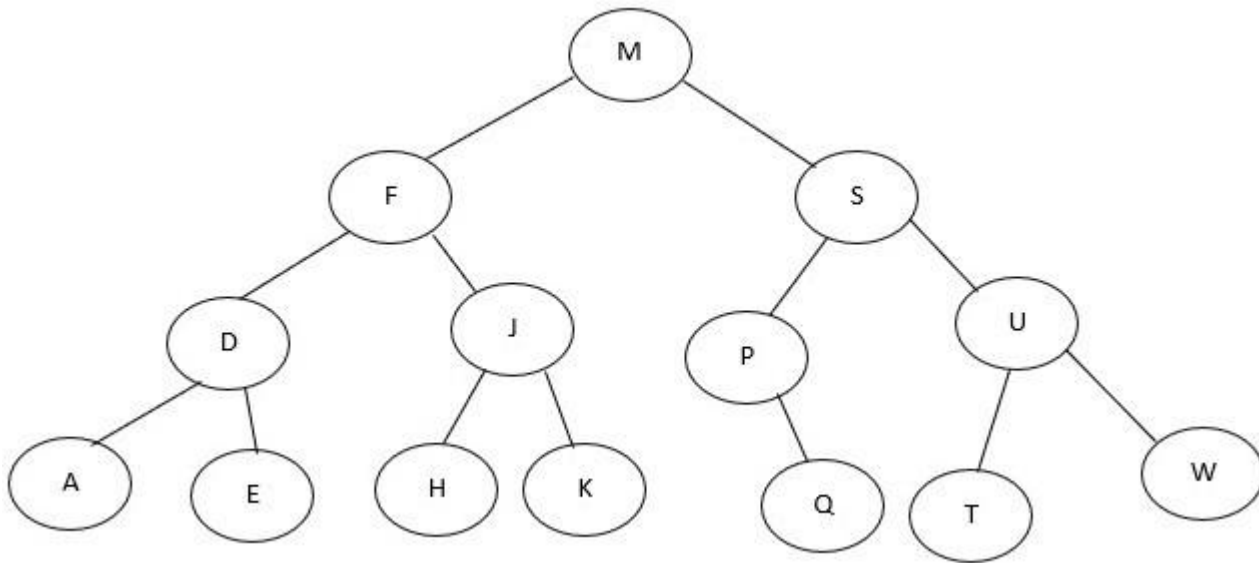
Resposta:

```
int verificaBin(Arvore *a){
    return(verificaBin_rec(a->raiz));
}

int verificaBin_rec(No *p){
    No *q=p->esq; No *r=p->dir;
    if(p==NULL)
        return 0;
    if((q->info.chave<p->info.chave)&&( r->info.chave.p->info.chave))
        return 1;
    if((q->info.chave<p->info.chave) || ( r->info.chave.p->info.chave))
        return 0;
    return((verificaBin_rec(p->esq))&&(verificaBin_rec(p->dir))));
}
```

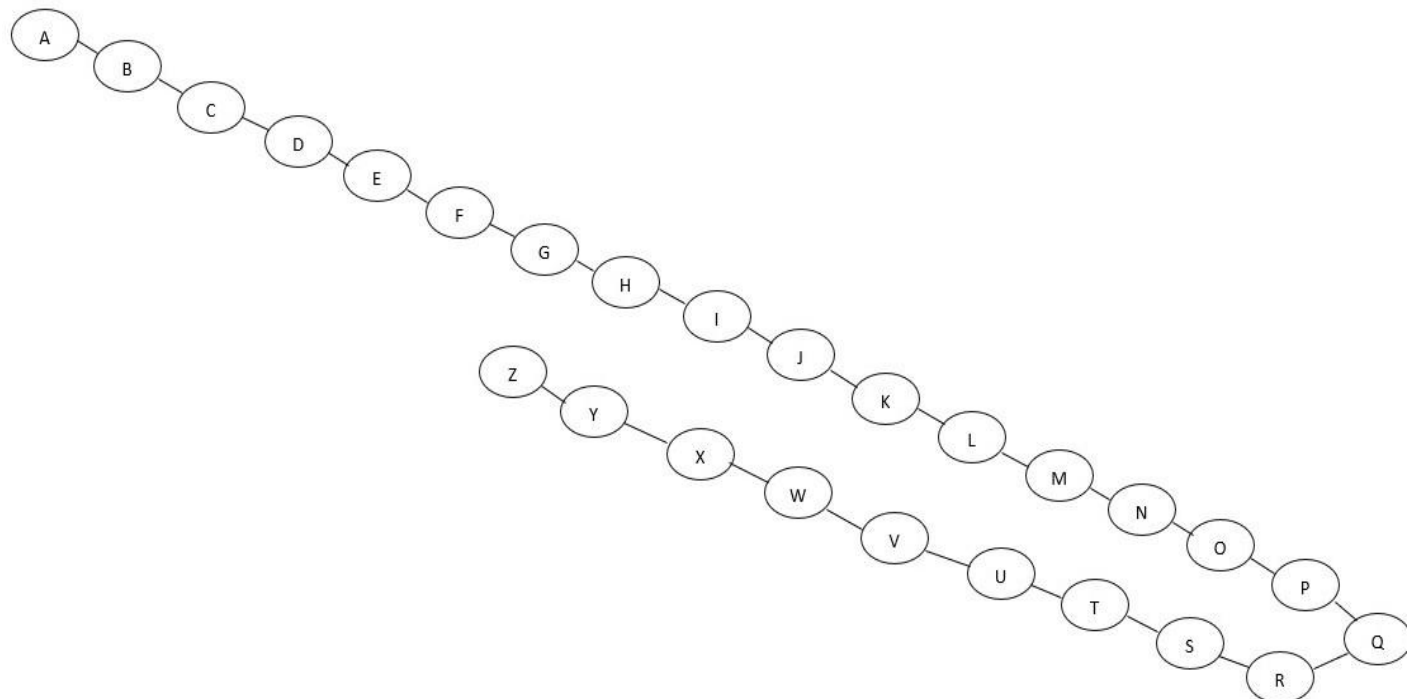
5- Dada uma ABB inicialmente vazia, insira (E DESENHE) os seguintes elementos (nessa ordem): M, F, S, D, J, P, U, A,E, H, Q, T, W, K.

Resposta:



6- Dada uma ABB inicialmente vazia, insira (E DESENHE) os seguintes elementos (nessa ordem): A, B, C, D, E, F, G, H,I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

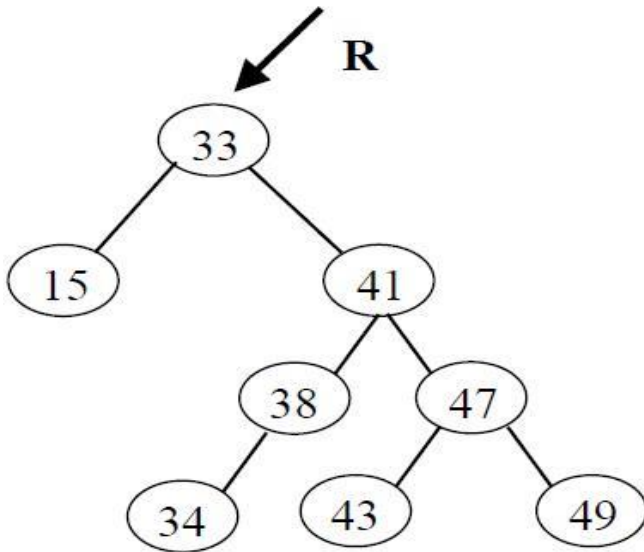
Resposta:



7- Implemente um TAD ABB.

Resposta:

8- Descreva a ordem de visita para um percurso em pré-ordem na árvore abaixo.

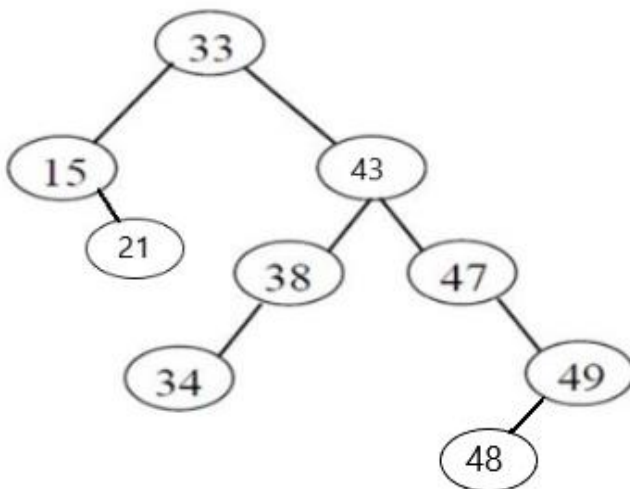


Resposta:

33-15-41-38-34-47-43-49

9- Considerando que a árvore acima represente uma ABB (considerando também a aplicação dos algoritmos para ABB estudados na disciplina), desenhe a nova árvore que será obtida após a realização das seguintes operações: inserir 21, remover 41, inserir 48.

Resposta:



33-15-21-43-38-34-47-49

10- Qual a diferença de uma ABB para uma AVL?

Resposta:

A árvore AVL é obrigatório que ela esteja balanceada enquanto que a binária não.

11- Esboce os casos básicos de inserção de elementos em uma AVL. Que casos podem acontecer?

Resposta:

12- As afirmações abaixo são verdadeiras ou falsas?

- i) São necessárias menos operações para inserir os elementos de um vetor ordenado em uma AVL (percorrendo-se o vetor da esquerda para a direita).
- ii) Para qualquer conjunto de dados, AVL é sempre mais eficiente do que uma ABB.
- iii) Busca binária e busca em uma ABB são igualmente eficientes no pior caso.

Resposta:

A opção 1 e 2

13- O que difere na implementação da função de busca de uma ABB e de uma AVL?

Resposta:

Na implementação da árvore binária de busca não necessário fazer o fator de balanceamento para verificar se está balanceada o qual acaba influenciando no tempo de busca.

14 -

(a) Insira os elementos a seguir em uma AVL (nessa ordem): 1, 2, 3, 4, 5, 6, 7, 8, 10, 15, 16, 17, 18, 20, 30, 29, 28, 27, 9, 12, 11, 14, 13, 1.

(b) Remova da AVL construída no item anterior os elementos 1, 7, 15, 17, 30, 9, 13 e 18 (nessa ordem).

Resposta: