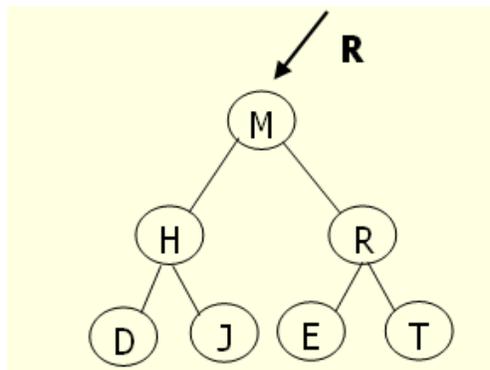


Lista de Exercícios

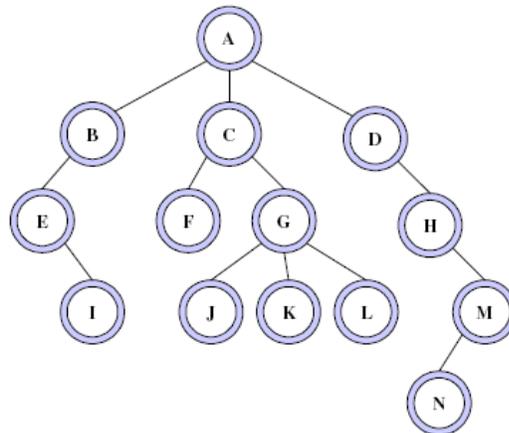
Árvore, Árvores Binárias, Árvores Binárias de Busca, Árvores AVL

Árvores

1. Dada a árvore abaixo, indique
os nós folha
o grau da árvore
o altura da árvore
os descendentes do nó H



2. Quantas árvores ordenadas existem com 3 nós?
3. Responda rápido:
 - a) Num diagrama convencional de árvore (raiz no topo), se o nó X tem nível maior que o nó Y, então X aparece abaixo de Y no diagrama?
 - b) Se o nó A tem 3 irmãos e B é pai de A, qual o grau de B?
4. Represente a árvore abaixo nas demais representações conhecidas (paragrafação, parênteses aninhados, diagramas de Venn).



5. Desenhe as árvores que correspondem às seguintes expressões aritméticas:

- a) $2 * (a - b / c)$;
- b) $a + b + 5 * c$.

Árvores Binárias

6. Diga, para cada uma das árvores binárias abaixo, se são balanceadas, perfeitamente balanceadas ou nenhum dos casos ou ambos e liste seus nós em (i) pré-ordem, (ii) in-ordem e (iii) pós-ordem:

- a) (1 (2 (4) (5)) (3 (6) (7)));
- b) (A (B (D (F)) (E)) (C (G (H))))).

7. Modifique o algoritmo não-recursivo de percurso de árvore binária em pré-ordem, de forma que a pilha lá usada seja encadeada. Há alguma vantagem nisso?

8. Uma quarta maneira de percorrer uma árvore binária é conseguida por meio dos quatro passos a seguir:

- (i) se árvore vazia, então fim;
- (ii) visite a raiz;
- (iii) percorra a sub-árvore direita pelo mesmo método;
- (iv) percorra a sub-árvore esquerda pelo mesmo método.

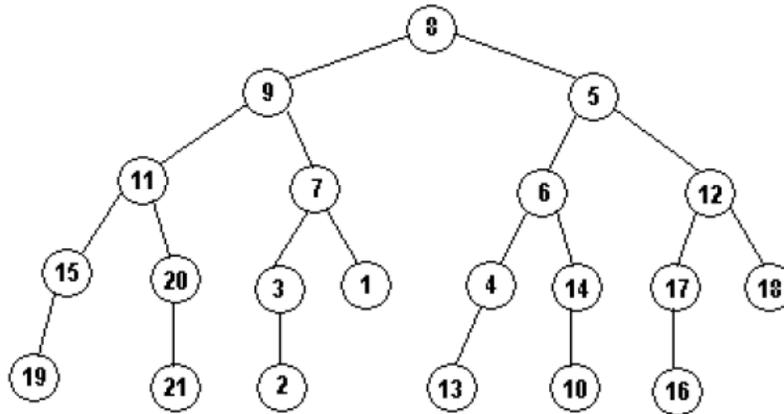
Essa nova ordem tem alguma relação com alguma das 3 ordens discutidas anteriormente? Quais os percursos resultantes da aplicação desse método às árvores do exercício 6?

9. Mostre que, se conhecermos a seqüência de nós em pré-ordem e in-ordem de uma árvore binária, podemos descobrir qual é a estrutura dessa árvore. Dê um exemplo.

10. Encontre todas as árvores binárias cujos nós aparecem exatamente na mesma seqüência em:

- a) pré- e in-ordem;
- b) pré- e pós-ordem;
- c) in- e pós-ordem.

11. Quais são as seqüências de nós encontradas ao atravessar a árvore abaixo em in-ordem, pré-ordem e pós-ordem?



12. Desenhe a árvore binária correspondente às seguintes seqüências em pré-ordem e inordem:

[1 2 3 4 5 6 7 8 9] e [3 2 6 5 4 1 7 8 9], respectivamente.

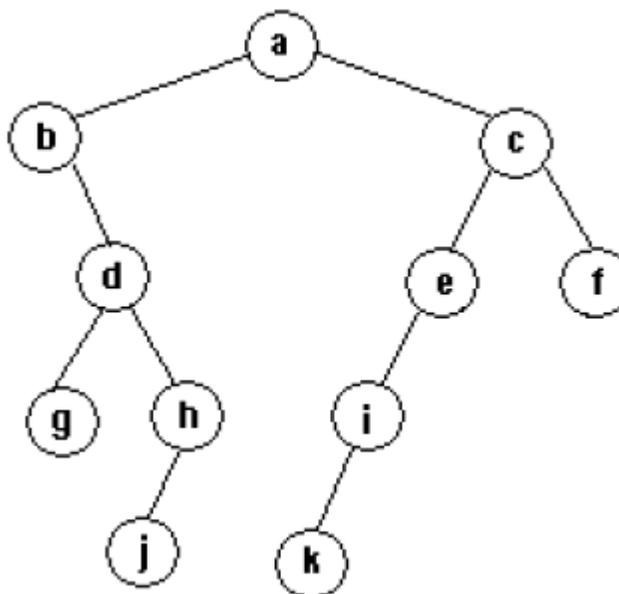
13. Escrever o algoritmo de visita em pré-ordem utilizando alocação dinâmica, mas sem

utilizar procedimentos recursivos. Utilizar **pilha** para saber o endereço da subárvore que resta à direita.

- processar raiz A
- guardar A para poder acessar C depois
- passa à B e processa essa subárvore
- idem para D
- retorna B (topo da pilha) para acessar D que é a subárvore esquerda

14. Você pode deduzir alguma relação entre a travessia pós-ordem em uma árvore binária e a travessia pré-ordem de sua imagem no espelho?

15. Determine a travessia pré, in e pós-ordem da árvore abaixo:



- 16.** Escreva uma função recursiva que conte o número de nós de uma árvore binária.
- 17.** Escreva uma função recursiva que verifique se uma árvore binária está balanceada.
- 18.** Considerando uma árvore de busca com n nós, qual é a relação entre o número de comparações (entre a chave procurada e chaves em nós) e a altura da árvore?
- 19.** Compute o número de nós ancestrais em uma árvore binária para um dado nó a um nível K .
- 20.** Liste 3 aplicações fora da ciência da computação onde a estrutura de árvores é útil. Para cada aplicação, desenhe a árvore típica, rotulando cada nó em termo das variáveis relevantes para a área de aplicação.

21. Escreva o procedimento cópia:

```
procedure cópia (t: tree; var c: tree);
(* cria uma árvore, c, que é a mesma de t *)
```

22. Escreva o procedimento espelho:

```
procedure espelho(t: tree; var e: tree);
(* cria uma árvore, e, que é a imagem no espelho de t *)
```

23. Considere uma árvore binária que representa uma expressão aritmética, definida pela estrutura:

```
type ponto-no = ^ no;
no = record
  esq, dir, pont-no;
case boolean of
  fase: (info-op: char);
  breve: (infor-num: real)
end;
```

Assuma que os únicos operadores usados são os operadores binários $*$, $+$, $-$, $/$. A função a seguir pretende calcular, recursivamente, o valor da expressão representada por uma árvore binária. Desenvolva os trechos (a) e (b).

```
function avalia (raiz: pont-no) : real;
var result1, result2 : real;
begin
  if (a) raiz é operando then
    avalia := raiz^.info-num
  else /* Raiz é operador */
    (b) cálculo de avalia neste ponto
  end;
```

Árvores Binárias de Busca

24. Num sistema de arquivos, um catálogo de todos os arquivos é organizado como uma árvore de busca binária. Cada nó denota um arquivo e especifica seu nome e, entre outras coisas, a data de seu último acesso, codificada como um inteiro. Escreva um programa que percorra a árvore e apague todos os arquivos cujos últimos acessos tenham sido anteriores a uma certa data. As chaves do catálogo são os nomes dos arquivos.

25. Numa árvore binária de busca a freqüência de acesso de cada elemento é medida empiricamente, atribuindo-se a cada nó um número de acessos. A cada certos intervalos de tempo, a organização da árvore é atualizada, percorrendo-se a árvore e gerando-se uma nova árvore usando o procedimento de busca com inserção, inserindo as chaves em ordem decrescente de sua freqüência de acesso. Escreva um programa que realize esta reorganização.

26. Faça uma sub-rotina (decida se procedimento ou função) para verificar se duas arvores binarias de busca sao similares. 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

27. 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 sub-rotina que verifica se duas ABBs são iguais.

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

29. Implemente uma sub-rotina para verificar se uma árvore binária é uma ABB.

30. 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

31. 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. O que se pode observar?

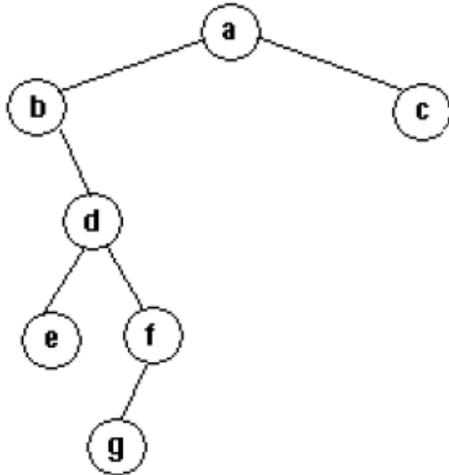
Árvores AVL

32. Defina árvore AVL.

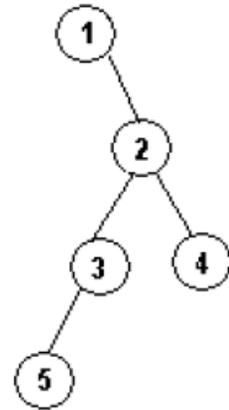
33. Escreva um procedimento que verifique se uma árvore é AVL.

34. Dada as seguintes árvores binárias abaixo, indique os passos para torná-las uma árvore binária balanceada (AVL).

(a)



(b)



35. Insira os números 35, 39, 51, 20, 13, 28, 22, 32, 25, 33 (nesta ordem) em uma árvore AVL.

36. Dê um exemplo de inserção de um elemento em uma árvore AVL que cause rearranjo da estrutura da árvore.

37. Dê um exemplo de remoção de um elemento de uma árvore AVL que cause rearranjo da estrutura da árvore.

38. Por que nos damos ao trabalho de procurar trabalhar com árvores binárias balanceadas? Justifique.