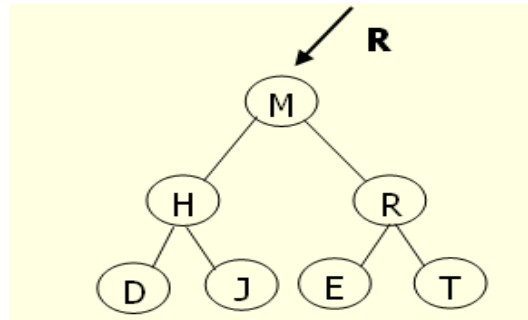
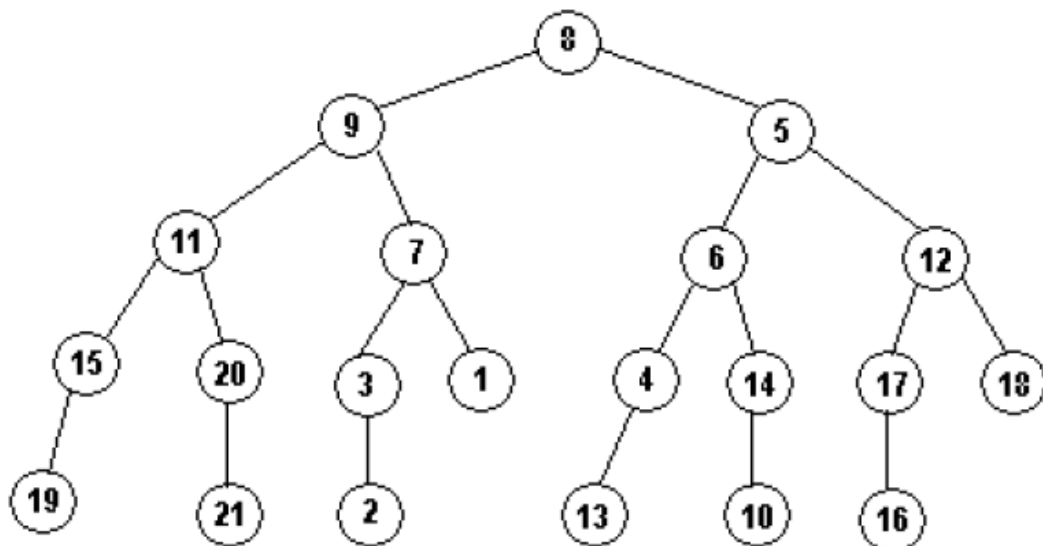


Lista de Exercícios de Estrutura de Dados

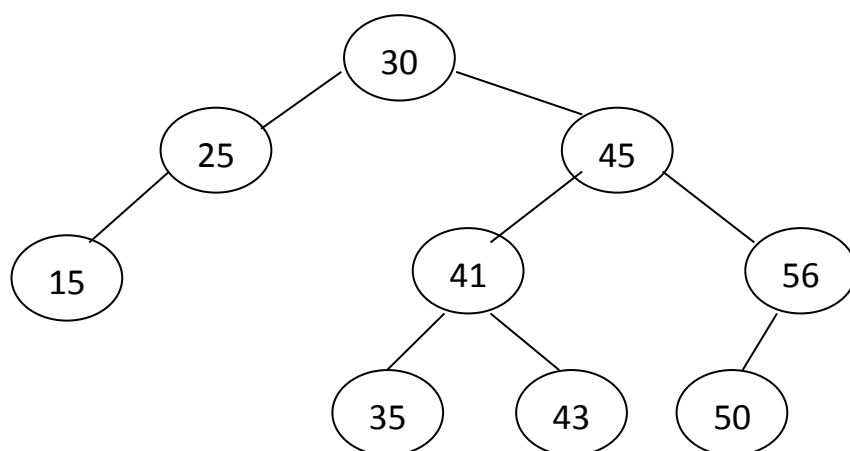
- 1) Dada a árvore abaixo, indique
 - a) os nós folha
 - b) o grau da árvore
 - c) a altura da árvore
 - d) os descendentes do nó H



- 2) 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))))).
- 3) Desenhe as árvores que correspondem às seguintes expressões aritméticas:
 - a) $2 * (a - b / c)$;
 - b) $a + b + 5 * c$.
- 4) Quais são as sequências de nós encontradas ao atravessar a árvore abaixo em in-ordem, pré-ordem e pós-ordem?



5) Considere a árvore AVL a seguir:



a) Realize, na árvore acima, as inserções das seguintes chaves 49, 60, 65, e em seguida a remoção das chaves 45 e 41, escolhendo necessariamente imediatamente precedente para a posição da chave removida. Mostre todas as rotações e o formato da árvore após cada operação.

b) Seja **q** um nó recém inserido e **p** o seu ancestral mais próximo que se tornou desregulado. Quais os possíveis valores para o fator de balanço de **p** após a inserção? Examinar o fator de balanço de **p** é suficiente para concluir se a inserção foi à esquerda ou a direita de **p**? Por que?

6) Desenhe a árvore binária correspondente às seguintes sequências em pré-ordem e in-ordem: [1 2 3 4 5 6 7 8 9] e [3 2 6 5 4 1 7 8 9], respectivamente.

7) Analise uma árvore T que armazena 100.000 itens. Quais são o pior e o melhor casos em relação à altura de T das seguintes árvores:

- T é uma árvore binária de pesquisa (ABP);
- T é uma árvore AVL.

8) Escreva uma função recursiva que conte o número de nós de uma árvore binária. Insira os números 35, 39, 51, 20, 13, 28, 22, 32, 25, 33 (nesta ordem) em uma árvore AVL. Escreva um procedimento que verifique se uma árvore é AVL.

9) 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.

void reorganiza (ARVORE_ABB *arv)

utilize as estruturas de dados visto em sala de aula.

10) Numa árvore binária de busca a frequência 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 frequência de acesso. Escreva um programa que realize esta reorganização.

11) Faça uma sub-rotina (decida se procedimento ou função) para verificar se duas árvores binárias de busca são 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 sub-árvores esquerdas são similares e suas sub-árvores direitas também são similares

12) Para todas as árvores vista em sala de aula (ABB, Heap Mínimo, AVL, Vermelho e Preto), inicialmente vazia, realize a inserção da seguinte sequência de chaves:

99, 44, 71, 80, 74, 63, 59, 120, 98, 150.

Redesenhe a árvore a cada inserção. Indique para cada rotação feita, o nome da rotação e o nó desbalanceado. Indique as árvores resultantes da exclusão dos nós 59 e 63.

13) Sem utilizar estruturas auxiliares (pilhas, filas, ...) escreva em C uma função não recursiva para determinar a altura de uma árvore AVL, visitando o menor número de nós possível. Use a estrutura:

```
typedef struct No {  
    int chave;  
    int bal; /* fator de balanço: hdir - hesq */  
    t_no *esq, *dir;  
}; tNO;
```

A função deve receber como parâmetro o endereço do nó raiz da árvore e retornar o valor da altura

14) Seja uma árvore AVL T . Considere a inserção de um nó q em T , que tornou T desregulada. Seja p o nó desregulado mais próximo das folhas.

a- Qual o valor exato de $|he(p) - hd(p)|$? Por que não pode ser nem mais nem menos?

b- Supondo $hd(p) > he(p)$ então existe um filho direito u de p . Por que necessariamente temos $|hd(u) - he(u)| = 1$? Por que não pode ser 2 ? Por que não pode ser 0 ?

c- De acordo com o item b, quando $hd(p) > he(p)$ existem dois sub-casos a serem considerados: $he(u) = hd(u) + 1$ ou $hd(u) = he(u) + 1$. Para cada um dos sub-casos acima, apresente a transformação que regula p (diga qual e apresente um esquema). Mostre que realmente todos os nós originalmente em T ficam regulados (através da análise das alturas das subárvores).

d- Por que a regulagem de p (nó desregulado mais próximo das folhas) regula toda a árvore ?

15) Escreva, em C, uma função recursiva que permuta as subárvores esquerda e direita de todos os nós de uma árvore binária. A função deve receber como parâmetro o endereço do nó raiz da subárvore a ser processada. Explícite também a chamada externa

16) Utilizando uma árvore binária para representar uma expressão aritmética.

a. Represente, em uma árvore binária, a seguinte expressão aritmética, respeitando a precedência usual dos operadores: $2 * (5 - 3) + (2 + 6) / 4$. As folhas devem corresponder aos operandos e os nós interiores aos operadores. Não represente os parênteses.

- b. Defina, em C, a estrutura do nó da árvore binária utilizada para representar expressões. Assuma que os operandos são inteiros positivos e que apenas os operadores binários +, -, * e / são aceitos. A estrutura deve possuir o menor número possível de campos.
- c. Escreva em C uma função recursiva que recebe como parâmetro o endereço do nó raiz de uma árvore binária representando uma expressão e retorna como resultado o valor obtido na avaliação da expressão. Utilize a estrutura do nó definida no item c.
- d. Qual o percurso implementado pela função escrita no item anterior? O que realiza o procedimento de visita ?

17) Seja $dh(t) = |h_e - h_d|$ o módulo da diferença entre as alturas das sub-árvores esquerda h_e e direita h_d de uma árvore binária t , sendo $dh(t) = 0$ quando $t == \text{NULL}$. A função $dh_max(t)$ deve calcular o valor máximo de dh para a árvore t . Implemente o código da função recursiva auxiliar $aux_dh_max(t)$, que, em um mesmo percurso, calcula a altura da árvore t (armazenando o resultado em $*h$) e retorna o dh máximo encontrado. Você pode utilizar as funções $\max(a,b)$ (retorna o máximo entre a e b) e $\text{abs}(a)$ (retorna o valor absoluto de a).

18) Escreva um procedimento para multiplicar por X (passado como parâmetro) todos os elementos de uma determinada coluna (também parâmetro) de uma matriz esparsa com 1000 linhas e 10000 colunas, contendo inteiros. Realizar todas as substituições considerando que a matriz esparsa está representada pelo método de lista cruzada.

19) Escreva um procedimento para somar duas matrizes esparsa com 1000 linhas e 10000 colunas, contendo inteiros. Realizar todas as substituições considerando que a matriz esparsa está representada pelo método de lista cruzada.

20) Construir uma árvore vermelho -preto sabendo que:

- Os valores das chaves são: 3, 4, 5, 7, 8, 10, 13, 15
- A raiz é o nó com chave 10
- O nó com chave 10 é o pai do nó com chave 7
- Os filhos do nó com chave 7 são os nós com chaves 4 e 8
- Os filhos do nó com chave 4 são os nós com chaves 3 e 5
- Os nós com chaves 7 e 13 são vermelhos
- Observação: Usar a representação de uma árvore vermelho-preto que omita a sentinela $\text{nil}[T]$.

21) Considerando a árvore construída na questão 19, faça a remoção das na seguinte ordem

10, 3, 7, 5, 13, 4, 15, 8, 13

22) Construa um heap de m_aximo a partir de um vetor contendo os elementos 5, 3, 17, 10, 84, 19, 6, 22, 9.

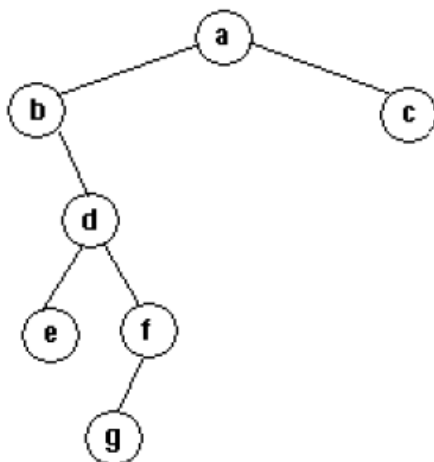
23) Escreva uma função recursiva `Altura(ArvBin *p)` que recebe um apontador para uma árvore binária e determina sua altura.

24) Escreva uma função recursiva que conta o número de nós em uma árvore binária que possuem pelo menos um filho.

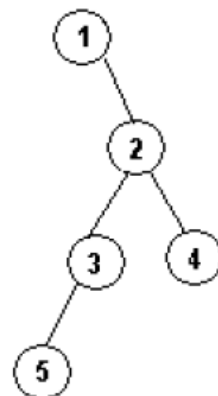
25) Escreva um função que retorna 1 caso o heap passado como argumento respeite as restrições de heap de máximo (para todo elemento que tenha pelo menos um filho, seu valor associado deve ser maior do que o do(s) seu(s) filho(s)) e 0 caso contrário.

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

(a)



(b)



Atenção: os exercícios foram adaptados de diferentes fontes: provas, livros: Cormen et. al, Ziviani et.atl.