



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

Disciplina: Estrutura de Dados	Turma: EC3/CC4	Data: 03/06/15	Nota:
Professor: Renato E. N. de Moraes	Semestre: 2015-1	Valor: 0,0 pt	
Aluno:	Lista de Exercícios 04		

1. Dado o conjunto de valores $C = 1, 4, 7, 9, 16, 23, 25$. Construa árvores binárias de pesquisa com alturas (número máximo de arestas entre a raiz e um nó folha) iguais a 2, 3, 4, 5 e 6.
2. Escreva uma função que calcule o número de células de uma árvore binária.
3. Escreva uma função que percorre uma árvore binária para determinar sua altura.
4. Escreva uma função recursiva que verifique se uma árvore binária está balanceada.
5. Escreva uma função de pesquisa em árvores binárias que conta quantas vezes uma chave k aparece na árvore.
6. Desenhe a árvore binária correspondente às seguintes seqüências em pré- ordem e em-ordem: [1 2 3 4 5 6 7 8 9] e [3 2 6 5 4 1 7 8 9], respectivamente.
7. Escreva a versão iterativa para a função de pesquisa em árvores binárias de pesquisa pesquisa_ABP(x, k), que encontra uma chave k a partir da raiz x .
8. Compute o número de nós ancestrais em uma árvore binária para um dado nó.
9. Numa árvore binária de busca (ABB) 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.
10. Escreva uma função que calcule o predecessor de uma chave em árvores binárias de pesquisa.
11. Escreva uma função que calcule o predecessor de uma chave em árvores binárias (não necessariamente de pesquisa).
12. 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
13. 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?
14. 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.
15. Implemente um procedimento para verificar se uma árvore binária é uma ABB.
16. Escreva um procedimento que verifique se uma árvore binária é AVL.



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

17. Mostre, passo a passo, a construção de uma árvore AVL a partir dos seguintes valores de chave, na ordem especificada, realizando as operações de rotação apropriadas quando necessário. Sequência: 14, 6, 4, 2, 16, 20, 12, 13.
18. Suponha o caso do nodo a ser removido de uma árvore binária de pesquisa ser substituído pelo seu sucessor:
 - (a) Como podemos garantir que sempre existirá um sucessor?
 - (b) Note que o algoritmo de remoção não contém chamadas recursivas. Como podemos garantir que o sucessor do nodo a ser removido também não contém dois filhos?
 - (c) Ao invés de usar o sucessor, poderíamos usar o predecessor?
19. Percorrer a árvore T da Figura 1 usando o percurso POS-ORDEM tal que o nó visitado seja inserido em uma árvore AVL. Mostre a situação da árvore AVL após cada inserção e, quando necessário a(s) rotação(ões) aplicadas. Mostre também o fator de balanceamento de cada nó.
20. Para a árvore construída no exercício anterior, esvazie a árvore, mantendo as propriedades AVL, removendo os nós na sequência dada pelo percurso PRE-ORDEM da árvore da Figura 1.

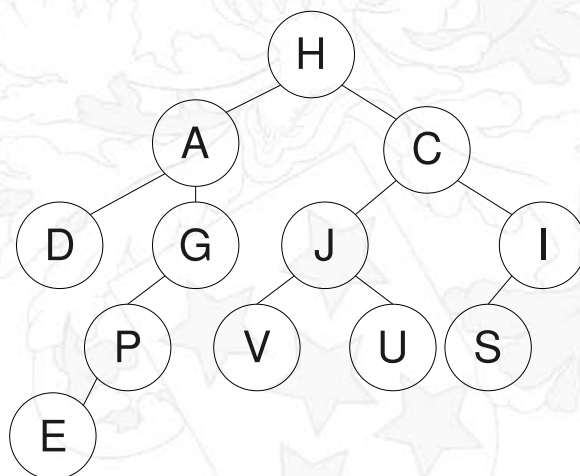


Figura 1: Árvore T , onde $\text{CONTEUDO}(\text{RAIZ}(T)) = H$.

21. Considere uma árvore binária de busca que armazena nomes (cadeias de caracteres). O tipo que representa um nó da árvore é dado por:

```

struct arv {
    char nome[81];
    struct arv* esq;
    struct arv* dir;
};

typedef struct arv Arv;
  
```



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

Escreva uma função recursiva que insira um novo nome na árvore de busca. Se o nome já existir na estrutura, a árvore não deve ser alterada. A função recebe como parâmetros a raiz da árvore e a cadeia de caracteres a ser inserida, e deve ter como retorno o valor atualizado da raiz. O protótipo dessa função deve ser: `Arv* insere (Arv* a, char* s);`

Obs: Para simplificar, considere que os nomes são compostos apenas por letras minúsculas.

Protótipos de funções que podem ser úteis: `int strlen (char* s); int strcmp (char* s, char *t); char* strcpy (char* destino, char* fonte); char* strcat (char* destino, char* fonte);`

22. Considere uma árvore binária que armazena valores inteiros. O tipo que representa um nó da árvore é dado por:

```
struct arv {  
    int val;  
    struct arv* esq;  
    struct arv* dir;  
};  
typedef struct arv Arv;
```

Pede-se

- (a) Escreva uma função que, dado o ponteiro para o nó raiz, calcule e retorne a altura da árvore. O protótipo dessa função deve ser: `int altura (Arv* a);`
 - (b) Considerando que essa árvore representa uma árvore binária de busca, escreva uma função para imprimir os valores associados aos nós da árvore em ordem decrescente. O protótipo dessa função é dado por: `void imprime (Arv* a);`
23. Por que nos damos ao trabalho de procurar trabalhar com árvores binárias balanceadas? Justifique.