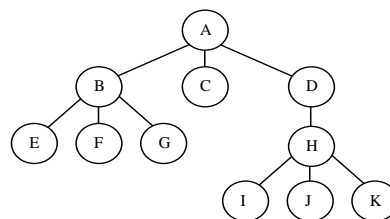


Segunda Lista de Exercícios

CC3651 – Estrutura de Dados

Prof. Rodrigo Filev Maia

1. O que é uma árvore binária e qual a vantagem (se houver alguma vantagem) na sua utilização com relação à uma árvore genérica.
2. Represente graficamente uma árvore qualquer e identifique: a raiz da árvore, as folhas, altura, nível e grau de cada nodo.
3. Escrever um algoritmo que, a partir de uma árvore binária de pesquisa já construída, exiba os seus elementos através das três formas de caminhamento. Para as três formas, apresentar os algoritmos recursivos e não-recursivos. Já existe um ponteiro chamado RAIZ que aponta para o elemento que está localizado na raiz da árvore.
4. Escreva um algoritmo para ler uma seqüência de valores inteiros e, para cada valor lido, incluí-lo em uma árvore binária de pesquisa. O valor 0 (zero) encerra a entrada de dados.
5. Representar, graficamente, a árvore genérica abaixo como árvore binária



6. Para a árvore binária obtida no exercício anterior, escreva as seqüências de nodos visitados após a execução dos percursos pré-ordem, inordem e pós-pós-ordem.
7. Escrever uma função recursiva que calcule a [altura](#) de uma árvore binária dada. A altura de uma árvore é igual ao máximo nível de seus nós
8. Dada uma árvore binária de pesquisa, onde cada nodo é constituído pelas seguintes informações: NOME, SEXO ('M' ou 'F'), IDADE e PESO. Sabendo que a árvore foi construída com a chave NOME e que já existe um ponteiro chamado RAIZ que aponta para o nodo raiz da árvore, construir algoritmos para

9. informar a quantidade de homens e a média de idade das mulheres;
10. pesquisar a idade e o peso de uma determinada pessoa.
11. Escreva um algoritmo que encontre o menor e o maior valor armazenados em uma árvore binária de pesquisa já construída.
12. Escreva um algoritmo para remoção de elementos em árvores binárias.
13. Defina em poucas palavras uma árvore balanceada e qual a finalidade de sua utilização.
14. Para os grafos abaixo, forneça: conjunto de nodos N e arcos A; exemplos de arcos incidentes (incidentes de um nodo, incidentes para um nodo e incidentes em um nodo); todos os nodos adjacentes; grau de cada nodo; todos os caminhos do nodo 1 para o nodo 3, com seus respectivos tamanhos; todos os ciclos e todos os laços. Adicionalmente, informe para cada um dos grafos se o mesmo é conectado ou fortemente conectado.
15. 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.
 - a. processar raiz A
 - b. guardar A para poder acessar C depois
 - c. passa à B e processa essa subárvore
 - d. idem para D
 - e. retorna B (topo da pilha) para acessar D que é a subárvore esquerda
16. Implemente uma função que retorne a quantidade de folhas de uma árvore binária.
17. Implemente uma função que compare se duas árvores binárias são iguais.
18. Escreva um algoritmo que encontre o menor e o maior valor armazenados em uma árvore binária de busca já construída.
19. Considere a seguinte função de hashing:

```
int hash(char *v, int M){  
    int h = 0, a = 128;  
    for (; *v != '\0'; v++) h = (a*h + *v) % M;  
    return h;  
}
```

20. Suponha que o tamanho M da tabela que está sendo usada é 1024. O usuário percebe que muitas colisões ocorrem quando esta função de hashing é utilizada para processar as palavras que ocorrem em um dado texto (isto é, o espalhamento esperado não ocorre). Você teria uma explicação para este fenômeno? Em particular, quando ocorre desta função devolver o mesmo valor para chaves distintas?
21. Suponha que estamos utilizando uma Tabela de Hashing com M entradas, com resolução de colisões por rehash linear (as M listas não são mantidas em ordem). Suponha que a função de hashing utilizada seja $h(\text{chave}) = 11k \bmod M$, onde chave seja uma letra do alfabeto, K a posição da letra no alfabeto (por exemplo, C corresponde a $k = 3$) e M o número de entradas na tabela. Suponha que desejamos inserir na tabela, inicialmente vazia, as seguintes chaves nesta ordem:

E A S Y Q U T I O N

onde $M = 16$. Ilustre as inserções das chaves na Tabela de Hashing.

22. Este é um exercício sobre filas de prioridade (heap). Na sequência

P R I O * R * * I * T * Y * * * Q U E * * * U * E

uma letra significa a inserção daquela letra em uma fila e um asterisco significa a remoção do item de maior prioridade desta fila. Qual é a sequência de chaves devolvida pelas 12 operações de remoção acima?

23. Uma companhia aérea tem 200 tripulantes que são identificados por um inteiro entre 1 e 200 (código de tripulante). Queremos uma tabela de hash que permita guardar a constituição da tripulação que segue num determinado avião. Deve depois permitir responder à questão: “O tripulante k viaja neste vôo?” (onde k é um código de tripulante). Se a resposta for afirmativa deve mostrar os seus dados. As estruturas que guardam a informação são do tipo seguinte:

```
struct{
    int codigo, idade;
    char nome[100];
};
```

Implemente uma tabela de hash com 3 posições, com resolução de colisões por encadeamento, para o problema do exercício anterior.

24. Escreva uma função que devolva a média de idades duma tripulação. Sendo necessária a consulta a todos os elementos armazenados, acha que a tabela de hash é uma boa escolha como estrutura de dados ?

25. Considere uma tabela de hash de tamanho $m = 1000$ e a função de hash $h(k) = [m(kA \bmod 1)]$, com $A = (\sqrt{5} - 1)/2$. Calcule os valores de hash das chaves 61, 62, 63, 64 e 65.
26. Uma função de hashing está sendo definida para mapear strings para a faixa de valores entre 0 e 63 da seguinte maneira: o resultado intermediário é a somatória dos valores ASCII de cada caráter não-nulo da string; esse resultado é dividido por 64 e o resto da divisão é o valor de hash (método da divisão inteira). Avalie o resultado dessa função para as seguintes strings:
- FEI
 - Centro Universitário da FEI
 - CC3651
 - Curso de Estrutura de Dados eh jóia!
27. Há colisões para as strings citadas acima, ou seja, strings diferentes são mapeadas para o mesmo valor entre a faixa de valores entre 0 e 63 ? observação – utilize o valor dos caracteres da tabela ASCII.
28. Mostre as árvores vermelho-preto que resultam após a inserção bem-sucedida das chaves 41,38, 31,12,19 e 8 em uma árvore vermelho-preto inicialmente vazia.
29. Desenhe uma árvore rubro-negra resultante da inserção dos valores do exercício anterior. Como a árvore se altera se for inserido o valor 36? O nó inserido é vermelho ou preto? Justifique sua resposta.