Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)
Professora: Juliana de Santi (jsanti@utfpr.edu.br)

Estruturas de dados II

Lista de exercícios

1) Baseado nas notas de aula e usando o código disponibilizado no Moodle, implemente as funções para realizar inserção, busca e remoção em uma árvore binária de busca (ABB).

```
Como exemplo, para construir a árvore:
          50
      30      90
  20    40        95
10     35 45

Faça:

int main () {
Arvore *a;
a = inserir (a, 50);
a = inserir (a, 30);
a = inserir (a, 90);
a = inserir (a, 20);
a = inserir (a, 40);
a = inserir (a, 95);
a = inserir (a, 10);
a = inserir (a, 35);
a = inserir (a, 45);

return 0;
}
```

2) Escreva uma função "min" que encontre (e imprima) uma chave mínima em uma árvore binária de busca. Escreva uma função "max" que encontre (e imprima) uma chave máxima. Utilize os seguintes protótipos para a sua função:

```
int min (Arvore *a);

int max (Arvore *a);
```

3) Escreva uma função recursiva que imprime uma árvore binária de busca em ordem decrescente. Utilize o seguinte protótipo para a sua função:

```
void imprime_decrescente (Arvore *a);


Exemplo:
         50
     30      90
  20   40       95
10   35  45

Saída: 95, 90, 50, 45, 40, 35, 30, 20, 10.
```

4) Escreva uma função que retorna o valor do maior caminho (soma dos valores dos nós) da raiz até algum nó folha da árvore. Utilize o seguinte protótipo para a sua função:

```
int maior_ramo (Arvore *a);

Por exemplo, a árvore:

         50
     30      90
  20   40       95
10   35  45

Tem os seguintes caminhos da raiz até as folhas:
  - 50, 30, 20, 10 = 110 (soma)
  - 50, 30, 40, 35 = 155 (soma)
  - 50, 30, 40, 45 = 165 (soma)
  - 50, 90, 95 = 235 (soma)
O valor do maior deles é: 235.
```

5) Escreva um programa que **(caso tenha problema na execução, reduza o tamanho do conjunto considerado)**:

a) produza 100000 números em ordem de 0 até 99999. Insira esses números em uma **árvore binária de busca** (ABB). Procure por um valor que não existe, por exemplo 100000. Quanto tempo a busca levou?

b) produza 100000 números aleatórios entre 0 e 99999. Insira esses números em uma **árvore binária de busca** (ABB). Procure por um valor que não existe, por exemplo 100000. Quanto tempo a busca levou?

Qual a explicação para a diferença de tempo para o item *a*) e o item *b*)?

Os exercícios a seguir (*) são exercícios extra, ou seja, não fazem parte da lista para pontuação. Não haverá lista-solução e o estudante que quiser resolve-los, deverá pensar a solução por conta própria.

Exercício *) Escreva uma função que retorna o ancestral comum de maior nível de dois nós fornecidos. Utilize o seguinte protótipo para a sua função:

```
int ancestral (Arvore *a, int e1, int e2);

Exemplo:

     20
  8      22
4   12
  10  14

ancestral (a, 4, 14) = 8.

ancestral (a, 4, 22) = 20.

ancestral (a, 10, 14) = 12.
```

Exercício *) Escreva uma função que transforma um vetor com elementos em ordem crescente em uma árvore de busca balanceada. Utilize o seguinte protótipo para a sua função:

```
Arvore* constroi_balanceado (int v[], int esq, int dir);

Exemplo:

vetor de entrada = [1, 2, 3, 4, 5, 6, 7]

Árvore resultado:
      4
  2       6
1   3   5   7
```

Exercício *) Para o conjunto de valores 1, 4, 5, 10, 16, 17, 21 qual (is) altura(s) para uma binária de busca não são possível(is)? Suponha que você deve inserir TODOS os valores e em qualquer ordem que achar necessário!

Exercício *) O custo das operações (busca, inserção e remoção) em uma árvore binária de busca (ABB) podem ser equivalentes ao custo destas operações em uma lista. É motivo para que isso ocorra:

a. Sua sub-árvore da direita ou sua sub-árvore da esquerda não atende as definições de ABB e, assim, aumenta a complexidade para fazer as operações em árvores desorganizadas.

b. Foram usadas chaves não únicas, o que deestabilizou a altura da árvore.

c. As chaves foram inseridas de tal forma que a altura da ABB é definida pela altura de uma árvore degenerada.

d. As chaves foram inseridas de tal forma que que altura da ABB é definida pela altura de uma árvore completa/cheia.

Exercício *) Em uma estrutura de árvore binária de busca foram inseridos os elementos 'h', 'a', 'b', 'c', 'i' e 'j', nesta sequência. O tamanho do caminho entre um nó qualquer da árvore e a raiz é dado pelo número de arestas neste caminho. Qual o tamanho do maior caminho na árvore, após a inserção dos dados acima?

Exercício *) Considerando uma árvore binária de busca completa com n nós e altura h. Verifique a veracidade das afirmações a seguir:

A) Nesta árvore existem 2^h folhas.

B) Nesta árvore existem $2 * 2^h$ possíveis posições para inserção de uma chave qualquer.