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 um programa que:

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*)?

4) 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.
```