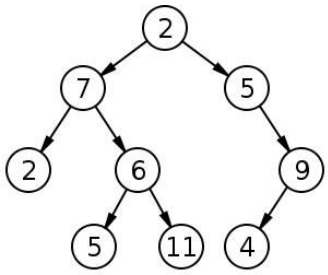
Instituto de Computação
Curso de Sistemas de Informação
TCC00171 – Estruturas de Dados I

3ª Lista de Exercícios – Árvores binárias – 31/10/2014

Nos exercícios 1 a 6, considere árvores binárias genéricas, que armazenam valores inteiros dispostos sem qualquer ordenação, e nos quais cada nó é do tipo *Arv*, descrito abaixo:

```
struct Arv {
   int info;
   struct Arv* esq;
   struct Arv* dir;
};
typedef struct Arv Arv;
```

1) Utilizando a função *arv_cria* apresentada em sala, escreva um programa que construa a árvore representada na figura abaixo.



2) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna o ponteiro para o nó da árvore que armazena o maior valor. Considere que essa árvore armazena valores positivo e negativos, mas não há valores repetidos. Caso a árvore seja vazia, a função deve retornar NULL. O protótipo da função é:

```
Arv* maior_valor(Arv* a);
```

3) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna o número total de folhas da árvore. O protótipo da função é:

```
int conta_folhas(Arv* a);
```

4) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna o número total de nós internos da árvore. O protótipo da função é:

```
int conta_internos(Arv* a);
```

5) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna a altura da árvore. O protótipo da função é:

```
int altura(Arv* a);
```

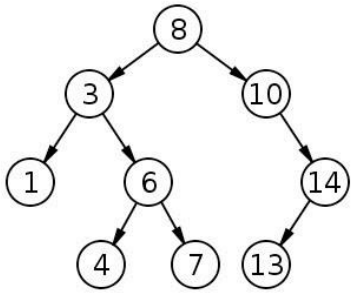
6) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e o valor de um nó e retorne o nível do nó na árvore. Caso o nó não exista, retorne -1. O protótipo da função é:

```
int nivel(Arv* a, int x);
```

Nos exercícios 7 a 13, considere árvores binárias de busca que armazenam valores inteiros, de forma que, para qualquer nó, o valor a ele associado é sempre maior ou igual que aqueles associados aos nós da sub-árvore da esquerda e menor àqueles associados aos nós da sub-árvore da direita, e cada nó é do tipo *Arv*, descrito abaixo:

```
struct Arv {
   int info;
   struct Arv* esq;
   struct Arv* dir;
};
typedef struct Arv Arv;
```

7) Utilizando a função *abb_insere* apresentada em sala, escreva um programa que construa a árvore representada na figura abaixo.



8) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca que armazena apenas valores positivos e retorna o maior valor armazenado na árvore. Caso a árvore seja vazia, a função deve retornar -1. O protótipo da função é:

```
int maior_valor(Arv* a);
```

9) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca e retorna o menor valor armazenado na árvore. Caso a árvore seja vazia, a função deve retornar NULL. O protótipo da função é:

```
Arv* menor_valor(Arv* a);
```

10) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca e um valor x, e retorna o número de nós que armazenam valores maiores do que x. O protótipo da função é:

```
int maiores_que_x(Arv* a, int x);
```

11) Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna 1, se esta for uma árvore binária de busca, ou 0, caso contrário. O protótipo da função é:

```
int verifica_abb(Arv* a);
```

12) Duas árvores binárias são similares se elas são vazias ou se ambas armazenam o mesmo valor em seu nó raiz, suas sub-árvores da esquerda são similares e suas sub-árvores da direita também são similares. Escreva uma função para determinar se duas árvores binárias são similares.

```
int similares(Arv* a1, Arv* a2);
```

13) Escreva uma função recursiva que recebe como parâmetros o ponteiro para o primeiro elemento de um vetor de inteiros ordenado e seu tamanho n (onde $n = 2^m - 1$ para $m > 0$) e retorne o ponteiro para uma árvore binária de busca completa, construída a partir dos valores contidos no vetor. A função deve ser executada em tempo linear, ou seja, deve ter complexidade O(n). O protótipo da função é:

```
Arv* cria_arv(int* v, int n);
```