P1 de INF1010 – EDA

1. Insira um conjunto de chaves numéricas em uma árvore AVL de forma que ocorra uma rotação dupla-esquerda, uma rotação dupla-direita, uma rotação simples à esquerda e uma rotação simples à direita, nesta ordem. Quando necessário, indique o nó desregulado e a rotação utilizada para regulá-lo. Redesenhe a árvore a cada passo.

2. Responda Certo ou Errado, justificando.

a. Qualquer que seja o número de chaves, é sempre possível construir com elas uma árvore binária completa.
b. Qualquer que seja o número de chaves, é sempre possível construir com elas uma árvore binária cheia.
c. Dada uma árvore binária com mais de 3 nós, é possível que um percurso em pre-ordem e um percurso em ordem simétrica visitem os nós na mesma ordem.
d. Toda árvore estritamente binária não-vazia possui número ímpar de nós.
e. Seja p o nó pai de um nó f em uma árvore binária, então:
   i. h (p) = h (f) + 1?
   ii. nível (p) = nível (f) +1 ?

   Obs: h (p) indica a altura do nó p e h (f) indica a altura do nó f.

3. Escreva em C ou em pseudocódigo uma função para determinar se uma árvore é binária de busca. Defina a estrutura de dados dos nós da árvore. A função deve receber como parâmetro o endereço do nó raiz da árvore e deve retornar TRUE se for ABB e FALSE, caso contrário. Use a seguinte estrutura:

```
struct no {
    void  *chave;
    struct no *esq, *dir;
};
```

4. Escreva em C ou em pseudocódigo, uma função mostracaminholongo que imprima todas as chaves dos nós do caminho mais longo da raiz até uma folha de uma árvore AVL. Se existirem vários caminhos de mesmo tamanho, a função pode imprimir qualquer um deles. Sua função pode ser recursiva ou não. Suponha a seguinte representação de árvore:

```
typedef struct avl Avl;
struct avl {int chave; int hesq; int hdir; Avl* esq; Avl* dir; };
```

E o seguinte protótipo da função:

```
void mostracaminholongo (Avl *r);
```

5. Qual a ordem da complexidade em tempo dos algoritmos A e B abaixo? Explique sua resposta.

```
A- int fat (int n) {
        if (n==0)
            return 1;
        return n* fat (n-1);
   }

B- for ( i=1; i < n; i << 1 ) {
        for ( j = n; j > 0; j /= 2 ) {
            for ( k = j; k < n; k += 2 ) {
                sum += (- j*k) * i/2;
            }
        }
   }
```

OBS: O operador << é o de deslocamento de bits à esquerda

6. Faça a representação binária da seguinte floresta de árvores n-árias: