

Exercício 04.

Pré-ordem:

- Visita o nó atual;
- Percorre recursivamente a sub-árvore esquerda do nó atual;
- Percorre recursivamente a sub-árvore direita do nó atual.

Listagem em pré-ordem:

A C B H J N E G D F I K M O P L

In-ordem:

- Percorre recursivamente a sub-árvore esquerda do nó atual;
- Visita o nó atual;
- Percorre recursivamente a sub-árvore direita do nó atual.

Listagem em in-ordem:

H J N B C G E A D O M P K I L F

Pós-ordem:

- Percorre recursivamente a sub-árvore esquerda do nó atual;
- Percorre recursivamente a sub-árvore direita do nó atual;
- Visita o nó atual.

Listagem em pós-ordem:

N J H B G E C O P M K L I F D A

Exercício 07.

A árvore estará rotacionada 90° no sentido anti-horário. Ou seja, deve-se imprimir partindo do elemento mais profundo e mais à direita possível, e acabando no elemento mais profundo mais à esquerda possível da árvore.

No percurso in-ordem, a ordem de visita dos nós é o inverso da que o problema requer. Portanto, invertendo-se a ordem de recursão (percorrendo-se recursivamente a sub-árvore direita antes da sub-árvore esquerda), obtém-se a ordem de impressão desejada.

O espaçamento utilizado para dividir as profundidades foi o de um *tab* (representado pela sequência de escape '\t'), que, na maioria dos editores de texto, corresponde à quatro espaços.

```
#ifndef IMPRIME_PROPORCIONAL_A_PROFUNDIDADE
#define IMPRIME_PROPORCIONAL_A_PROFUNDIDADE
#include <stdio.h>

typedef struct cel {
    int info;
    struct cel * esq;
    struct cel * dir;
} no;

/* valor inicial de "profundidade" deve ser 0 */
void imprimeProporcional(no *raiz, int profundidade)
{
    int i;

    if(raiz != NULL)
    {
        imprimeProporcional(raiz->dir, ++profundidade);
        printf("\n");

        for(i = 0; i < profundidade; ++i)
            printf("\t");

        printf("%d\n", raiz->info);

        imprimeProporcional(raiz->esq, profundidade);
    }
}

#endif
```