

Estrutura de Dados - MC202 A

1º Semestre de 2018

Tiago de Paula Alves - 187679

Lista 3 - Exercício 2

Aqui, foi usada a ideia de que em uma árvore binária de busca, todos os elementos à esquerda são menores que a raiz e todos à direita são maiores. Então, para fazer uma lista ordenada, basta fazer uma lista com as sub-árvores esquerda e direita e conectá-las com a raiz no meio.

```
1 p_no transforma_em_lista(p_no raiz) {
2     p_no esq_ini, esq_fim;
3     p_no dir_ini, dir_fim;
4
5     /* uma árvore vazia já é uma lista ordenada */
6     if (raiz == NULL) {
7         return NULL;
8     }
9
10    /* reordena a sub-árvore esquerda em uma lista */
11    esq_ini = transforma_em_lista(raiz->esq);
12    /* se a lista for vazia */
13    if (esq_ini == NULL) {
14        /* troca a lista pela raiz */
15        esq_ini = esq_fim = raiz;
16    } else {
17        /* senão, encontra o final da lista */
18        esq_fim = esq_ini->esq;
19    }
20
21    /* faz o mesmo com a direita */
22    dir_ini = transforma_em_lista(raiz->dir);
23    if (dir_ini == NULL) {
24        dir_ini = dir_fim = raiz;
25    } else {
26        dir_fim = dir_ini->esq;
27    }
28
29    /* conecta a lista esquerda na raiz */
30    esq_fim->dir = raiz;
31    raiz->esq = esq_fim;
32    /* e a raiz na lista direita */
33    dir_ini->esq = raiz;
34    raiz->dir = dir_ini;
35
36    /* resolve a circularidade da lista */
37    dir_fim->dir = esq_ini;
38    esq_ini->esq = dir_fim;
39
40    return esq_ini;
41 }
```