

Estrutura de Dados - MC202 A

1º Semestre de 2018

Tiago de Paula Alves - 187679

Lista 3 - Exercício 1

Para este código, usei o conceito de árvore binária cheia, que é uma árvore com todos os níveis cheios. Então, para que uma árvore seja cheia, suas sub-árvores também devem ser cheias e têm que ter a mesma profundidade. Com esse conceito, podemos dizer que uma árvore só pode ser completa se suas sub-árvores forem completas, sendo pelo menos uma cheia, e a direita ter profundidade igual a da esquerda ou reduzido de um, como visto abaixo.

```
1  /* Se for completa, retorna a profundidade da árvore + 1,  
2  * se for cheia, retorna o negativo disso,  
3  * senão, retorna 0.  
4  */  
5  int eh_completa(p_no raiz) {  
6      int prof_esq, prof_dir;  
7      bool esq_cheia, dir_cheia;  
8      bool cheia, completa;  
9  
10     /* árvore vazia é cheia com profundidade 0 */  
11     if (raiz == NULL) {  
12         return -1;  
13     }  
14  
15     /* análise da profundidade, da completude e da plenitude  
16     das sub-árvores esquerda e direita */  
17  
18     prof_esq = eh_completa(raiz->esq);  
19     if (prof_esq < 0) {  
20         esq_cheia = true;  
21         prof_esq = -prof_esq;  
22     } else if (prof_esq > 0) {  
23         esq_cheia = false;  
24     } else {  
25         return 0;  
26     }  
27  
28     prof_dir = eh_completa(raiz->dir);  
29     if (prof_dir < 0) {  
30         dir_cheia = true;  
31         prof_dir = -prof_dir;  
32     } else if (prof_dir > 0) {  
33         dir_cheia = false;  
34     } else {  
35         return 0;  
36     }  
37  
38     /* com isso, sabemos que as sub-árvores são completas e  
39     podemos analisar a completude da árvore principal */  
40  
41     cheia = esq_cheia && dir_cheia && prof_esq == prof_dir;  
42
```

```
43
44     completa = (esq_cheia && prof_esq == prof_dir)
45                 || (dir_cheia && prof_esq == prof_dir+1);
46
47     /* com base na plenitude e na completude,
48        retornamos o valor */
49
50     if (completa) {
51         return (cheia? -1 : 1) * (prof_esq + 1);
52     } else {
53         return 0;
54     }
55 }
```