

Ex 3.2

Testar se árvore é balanceada

10

int ébalanceada (BTree t)

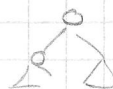
```

{
    int res = 1;
    if (!t) {
        if (altura(t->esp) == (altura(t->dir))) {
            res = min(ébalanceada(t->esp), ébalanceada(t->dir));
        }
        else res = 0;
    }
    return res;
}

```



$T(N) = 4$



$T(N) = 1 + \frac{T(N-1)}{2}$

Análise zumbado

Ficha 4

3.1

Melhor caso: constante

Pior caso: linear (n-1)

Caso médio: constante (2)

1	0	1	0	1	1
---	---	---	---	---	---

$\frac{1}{64}$   $\frac{1}{32}$   $\frac{1}{16}$   $\frac{1}{8}$   $\frac{1}{4}$   $\frac{1}{2}$

valores jogados de incrementos

	i	S <sub>i</sub>	S <sub>i+1</sub>	c <sub>i</sub>	
operação número	1	0000	0001	1	1/1 das entradas tem um 1
	2	0001	0010	2	1/2 " " " " " 2 " "
	3	0010	0011	1	1/4 " " " " " 3 " "
	4	0011	0100	3	1/8 " " " " " 4 " "
	5	0100	0101	1	
	6	0101	0110	2	
	7	0110	0111	4	
	8	0111	1000	4	

$$\sum_{i=1}^n c_i = N \cdot \left( 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots \right)$$

$$= n \sum_{i=1}^{\log_2 n} \frac{1}{2^i} = n(2 - \frac{1}{n})$$

estado estabilizado:

$c_i = 2 - \frac{1}{n}$

Método estatístico (cada operação n por um c)

$c_i = 2$  : Estado estabilizado por muitos 11 pouco zeros

i	c <sub>i</sub>	Saldo final
1	1	1
2	2	1
3	1	2
4	3	3-4 = -1
5	1	2

$$\text{Saldo} = \sum_{i=1}^n c_i - \sum_{i=1}^n c_i$$

$$S_i \geq 0$$