

Aula 15

AVL: Algoritmo Remover

Estruturas de Dados Avançadas

Professor Eurinardo Rodrigues Costa
Universidade Federal do Ceará
Campus Russas

2021.1

Sumário

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

1 Aulas Passadas

2 Rotações

3 Árvore Binária de Busca AVL

- Algoritmo Remover

Aulas Passadas

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Definição (Árvore Binária de Busca AVL)

ABB em que cada nó v temos que

$$-1 \leq h_D(v) - h_E(v) \leq 1$$

- $h_D(v)$ = altura da subárvore *direita* de v
- $h_E(v)$ = altura da subárvore *esquerda* de v

*neste caso dizemos que v é **regulado** e, em caso contrário, dizemos que v é **desregulado**.*

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 2: $\text{Caso1R}(pt, h)$

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 3: Caso1R(*pt*, *h*)

```
1  ptu ← pt.esq
2  se ptu.bal ≤ 0 então
3      | pt.esq ← ptu.dir
4      | ptu.dir ← pt
5      | pt ← ptu
6      | se ptu.bal = -1 então
7          | ptu.bal ← pt.dir.bal ← 0
8          | h ← V
9      | senão
10         | ptu.bal ← 1
11         | pt.dir.bal ← -1
12         | h ← F
13 senão
14     | ptv ← ptu.dir
15     | ptu.dir ← ptv.esq
16     | ptv.esq ← ptu
17     | pt.esq ← ptv.dir
18     | ptv.dir ← pt
19     | pt ← ptv
20     | caso ptv.bal seja
21         | -1 : ptu.bal ← 0; pt.dir.bal ← 1;
22         | 0 : ptu.bal ← 0; pt.dir.bal ← 0;
23         | 1 : ptu.bal ← -1; pt.dir.bal ← 0;
24     | pt.bal ← 0; h ← V;
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 5: $\text{balancear}(pt, R, h)$

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 6: $\text{balancear}(pt, R, h)$

1 **se** $h = V$ **então**

|

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 7: balancear(pt , R , h)

```
1 se  $h = V$  então  
2   se  $R = D$  então
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 8: $\text{balancear}(pt, R, h)$

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 9: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 10: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 11: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
5       0 :
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 12: balancear(*pt*, *R*, *h*)

```
1 se h = V então
2   se R = D então
3     caso pt.bal seja
4       1 : pt.bal ← 0;
5       0 : pt.bal ← -1;
```


Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 13: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
5       0 :  $pt.bal \leftarrow -1$ ;  $h \leftarrow F$ ;
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 14: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
5       0 :  $pt.bal \leftarrow -1$ ;  $h \leftarrow F$ ;
6       -1:
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 15: balancear(pt , R , h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
5       0 :  $pt.bal \leftarrow -1$ ;  $h \leftarrow F$ ;
6       -1 : Caso1R( $pt$ ,  $h$ );
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 16: balancear(pt, R, h)

```
1 se  $h = V$  então
2   se  $R = D$  então
3     caso  $pt.bal$  seja
4       1 :  $pt.bal \leftarrow 0$ ;
5       0 :  $pt.bal \leftarrow -1$ ;  $h \leftarrow F$ ;
6       -1 : Caso1R( $pt, h$ );
7   senão
8     caso  $pt.bal$  seja
9       -1 :  $pt.bal \leftarrow 0$ ;
10      0 :  $pt.bal \leftarrow +1$ ;  $h \leftarrow F$ ;
11      1 : Caso2R( $pt, h$ );
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 18: $\text{RemoverAVL}(x, pt, h)$

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remove

Algoritmo 19: $\text{RemoveAVL}(x, pt, h)$

1 **se** $pt = \lambda$ **então**

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 20: $\text{RemoverAVL}(x, pt, h)$

1 **se** $pt = \lambda$ **então** "elem. não existe"; $h \leftarrow F$

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 21: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 22: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então
```

```
    |
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 23: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4      $\text{RemoverAVL}(x, pt.esq, h)$ 
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 24: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$ 
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 25: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$   
6   senão
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 26: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$   
6   senão  
7     se  $x > pt.chave$  então  
       |
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 27: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$   
6   senão  
7     se  $x > pt.chave$  então  
8     |  $\text{RemoverAVL}(x, pt.dir, h)$ 
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 28: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$   
6   senão  
7     se  $x > pt.chave$  então  
8       |  $\text{RemoverAVL}(x, pt.dir, h)$   
9       |  $\text{balancear}(pt, D, h)$ 
```


Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 29: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10  senão
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 30: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$ 
```

Árvore Binária de Busca AVL

Algoritmo 31: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
        |
```

Árvore Binária de Busca AVL

Algoritmo 32: $\text{RemoverAVL}(x, pt, h)$

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     |  $\text{RemoverAVL}(x, pt.esq, h)$   
5     |  $\text{balancear}(pt, E, h)$   
6   senão  
7     se  $x > pt.chave$  então  
8       |  $\text{RemoverAVL}(x, pt.dir, h)$   
9       |  $\text{balancear}(pt, D, h)$   
10  senão  
11    |  $aux \leftarrow pt$   
12    | se  $pt.esq = \lambda$  então  
13      |  $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 33: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 34: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
          |
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 35: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 36: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17      senão
```


Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 37: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17      senão  
18         $s \leftarrow pt.dir$ 
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 38: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17      senão  
18         $s \leftarrow pt.dir$   
19        se  $s.esq = \lambda$  então
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 39: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17      senão  
18         $s \leftarrow pt.dir$   
19        se  $s.esq = \lambda$  então  
20           $s.esq \leftarrow pt.esq$   
21           $s.bal \leftarrow pt.bal$   
22           $pt \leftarrow s$   
23           $h \leftarrow V$ 
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 40: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17        senão  
18           $s \leftarrow pt.dir$   
19          se  $s.esq = \lambda$  então  
20             $s.esq \leftarrow pt.esq$   
21             $s.bal \leftarrow pt.bal$   
22             $pt \leftarrow s$   
23             $h \leftarrow V$   
24        senão
```

Árvore Binária de Busca AVL

Algoritmo 41: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17        senão  
18           $s \leftarrow pt.dir$   
19          se  $s.esq = \lambda$  então  
20             $s.esq \leftarrow pt.esq$   
21             $s.bal \leftarrow pt.bal$   
22             $pt \leftarrow s$   
23             $h \leftarrow V$   
24          senão  
25            enquanto  $s.esq \neq \lambda$  faça  
26               $PaiS \leftarrow s$   
27               $s \leftarrow s.esq$   
28            trocar( $pt, PaiS.esq$ )  
29            RemoverAVL( $x, pt.dir, h$ )
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 42: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17        senão  
18           $s \leftarrow pt.dir$   
19          se  $s.esq = \lambda$  então  
20             $s.esq \leftarrow pt.esq$   
21             $s.bal \leftarrow pt.bal$   
22             $pt \leftarrow s$   
23             $h \leftarrow V$   
24          senão  
25            enquanto  $s.esq \neq \lambda$  faça  
26               $PaiS \leftarrow s$   
27               $s \leftarrow s.esq$   
28            trocar( $pt, PaiS.esq$ )  
29            RemoverAVL( $x, pt.dir, h$ )  
30          balancear( $pt, D, h$ )
```

Árvore Binária de Busca AVL

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover

Algoritmo 43: RemoverAVL(x, pt, h)

```
1 se  $pt = \lambda$  então "elem. não existe";  $h \leftarrow F$ ;  
2 senão  
3   se  $x < pt.chave$  então  
4     RemoverAVL( $x, pt.esq, h$ )  
5     balancear( $pt, E, h$ )  
6   senão  
7     se  $x > pt.chave$  então  
8       RemoverAVL( $x, pt.dir, h$ )  
9       balancear( $pt, D, h$ )  
10    senão  
11       $aux \leftarrow pt$   
12      se  $pt.esq = \lambda$  então  
13         $pt \leftarrow pt.dir$ ;  $h \leftarrow V$ ;  
14      senão  
15        se  $pt.dir = \lambda$  então  
16           $pt \leftarrow pt.esq$ ;  $h \leftarrow V$ ;  
17        senão  
18           $s \leftarrow pt.dir$   
19          se  $s.esq = \lambda$  então  
20             $s.esq \leftarrow pt.esq$   
21             $s.bal \leftarrow pt.bal$   
22             $pt \leftarrow s$   
23             $h \leftarrow V$   
24          senão  
25            enquanto  $s.esq \neq \lambda$  faça  
26               $PaiS \leftarrow s$   
27               $s \leftarrow s.esq$   
28            trocar( $pt, PaiS.esq$ )  
29            RemoverAVL( $x, pt.dir, h$ )  
30          balancear( $pt, D, h$ )  
31      liberar( $aux$ )
```

Bibliografia

EDA - Aula 15

Prof.
Eurinardo

Aulas
Passadas

Rotações

Árvore Binária
de Busca AVL

Algoritmo Remover



SZWARCFITER , Jayme; MARKENZON, Lilian.
Estruturas de Dados e Seus Algoritmos.3a edição.
LTC, ano 2010. (ISBN 9788521617501).

Obrigado!