EDA - Aula 1

Prof. Eurinardo

Aulas Passada:

Rotações

Árvore Binária de Busca AVL

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. Eurinard

Aulas Passada:

Rotações

Árvore Binária de Busca AVL Algoritmo Remover Aulas Passadas

2 Rotações

- Árvore Binária de Busca AVL
 - Algoritmo Remover

Aulas Passadas

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

Aulas Passadas

FDA - 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.

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Arvore Binário de Busca AVI Algoritmo 2: Caso1R(pt, h)

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

```
Algoritmo 3: Caso1R(pt. h)
 1 ptu ← pt.esq
    se ptu.bal < 0 então
          pt.esq \leftarrow ptu.dir
          ptu.dir \leftarrow pt
          pt \leftarrow ptu
          se ptu.bal = -1 então
                 ptu.bal \leftarrow pt.dir.bal \leftarrow 0
 7
                 h \leftarrow V
 R
          senão
                 ptu.bal \leftarrow 1
10
                 pt.dir.bal \leftarrow -1
11
                 h \leftarrow F
12
13 senão
          ptv \leftarrow ptu.dir
14
          ptu.dir ← ptv.esq
15
          ptv.esq \leftarrow ptu
16
          pt.esq ← ptv.dir
17
          ptv.dir \leftarrow pt
18
          pt \leftarrow ptv
19
          caso ptv.bal seja
21
                 -1: ptu.bal \leftarrow 0: pt.dir.bal \leftarrow 1:
                 0: ptu.bal \leftarrow 0; pt.dir.bal \leftarrow 0;
22
                 1: ptu.bal \leftarrow -1: pt.dir.bal \leftarrow 0:
23
          pt.bal \leftarrow 0; h \leftarrow V;
24
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

EDA - Aula 15

Prof. Eurinardo

Aulas Passada

Rotações

Árvore Binári de Busca AV

Algoritmo Remove

Algoritmo 5: balancear(pt, R, h)

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Arvore Binária de Busca AVI

Algoritmo Remove

Algoritmo 6: balancear(pt, R, h)

 ${\bf 1} \ \ {\bf se} \ h = \ V \ {\bf ent \tilde{ao}}$

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Árvore Binária de Busca AVL

```
Algoritmo 7: balancear(pt, R, h)

1 se h = V então
2 se R = D então
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

```
Algoritmo 8: balancear(pt, R, h)
```

```
1 se h = V então

2 | se R = D então

3 | caso pt.bal seja
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

```
Algoritmo Remover
```

```
Algoritmo 9: balancear(pt, R, h)

se h = V então

se R = D então

caso pt.bal seja

1:
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

```
Algoritmo 10: balancear(pt, R, h)

se h = V então

se R = D então

caso pt.bal seja

1: pt.bal \leftarrow 0;
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binária de Busca AVI

```
Algoritmo Remove
```

```
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:
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

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

se h = V então

se R = D então

caso pt.bal seja

1: pt.bal \leftarrow 0;

0: pt.bal \leftarrow -1;
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

```
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;
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binária de Busca AVL

```
Algoritmo 14: balancear(pt, R, h)

1 se h = V então

2 se R = D então

2 caso pt.bal seja

4 | 1 : pt.bal \leftarrow 0;

5 | 0 : pt.bal \leftarrow -1; h \leftarrow F;

6 | -1:
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

```
Algoritmo 15: balancear(pt, R, h)
```

FDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVL

```
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;

-1: Caso1R(pt, h);

7 senão

8 caso pt.bal seja

9 1: pt.bal \leftarrow 0;

0 : pt.bal \leftarrow 0;

0 : pt.bal \leftarrow 1; h \leftarrow F;

11 Caso2R(pt, h);
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

Algoritmo 18: RemoverAVL(x, pt, h)

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Árvore Binár de Busca AV

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Árvore Binária de Busca AVL

Algoritmo Remover

Algoritmo 19: RemoverAVL(x, pt, h)

 ${\bf 1} \ \ {\bf se} \ pt = \lambda \ {\bf ent \tilde{ao}}$

EDA - Aula 15

Prof. Eurinardo

Aulas Passada:

Rotações

Árvore Binár de Busca AV

Algoritmo Remover

Algoritmo 20: RemoverAVL(x, pt, h)

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

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binár de Busca AV

Algoritmo Remover

Algoritmo 21: RemoverAVL(x, pt, h)

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

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binár de Busca AV

```
Algoritmo 22: RemoverAVL(x, pt, h)
```

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

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

Algoritmo Remover

Algoritmo 23: RemoverAVL(x, pt, h)

- 1 se $pt = \lambda$ então "elem. não existe"; $h \leftarrow F$;
- 2 senão
- se x < pt.chave então
 - RemoverAVL(x, pt.esq, h)

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
Algoritmo 24: RemoverAVL(x, pt, h)
```

- se $pt = \lambda$ então "elem. não existe"; $h \leftarrow F$; senão
- 2 senao
- se x < pt.chave então
- RemoverAVL(x, pt.esq, h)
 - balancear(pt, E, h)

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 25: 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)

balancear(pt, E, h)
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binár de Busca AV

```
Algoritmo 26: 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)

5 senão

7 se x < pt.chave então
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
Algoritmo 27: 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)
5 senão
7 ext{se } x > pt. chave então
8 RemoverAVL(x, pt. dir, h)
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
    Algoritmo 28: 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 se x > pt.chave então

    8 RemoverAVL(x, pt.dir, h)

    9 balancear(pt, D, h)
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
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 senão

5 senão

6 RemoverAVL(x, pt.esq, h)

6 senão

7 se x > pt. chave então

8 RemoverAVL(x, pt.dir, h)

9 balancear(pt, D, h)

5 senão
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVI

```
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

RemoverAVL(x, pt, esq, h)

balancear(pt, E, h)

senão

8 RemoverAVL(x, pt, dir, h)

balancear(pt, D, h)

senão

1 aux \leftarrow pt
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
      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

      8 RemoverAVL(x, pt. dir. h)

      9 balancear(pt. D, h)

      10 senão

      11 se pt. esq = \lambda então
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
Algoritmo 32: 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 senão

7 se x > pt.chave então

8 ex x > pt.chave então

RemoverAVL(x, pt.dir, h)
5 senão

10 se x > pt.chave então

RemoverAVL(x, pt.dir, h)
5 senão

11 se x > pt.chave então
12 se x > pt.chave então
13 x > pt.chave então
14 x > pt então
15 x > pt.chave então
16 x > pt.chave então
17 x > pt.chave então
18 x > pt.esq então
19 x > pt.esq então
19 x > pt.esq então
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binária de Busca AVI

```
Algoritmo 33: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
       se x < pt.chave então
            RemoverAVL(x, pt.esq, h)
            balancear(pt, E, h)
       senão
            se x > pt.chave então
                 RemoverAVL(x, pt.dir, h)
                 balancear(pt, D, h)
            senão
                 aux \leftarrow pt
                 se pt.esa = \lambda então
                     pt \leftarrow pt.dir; h \leftarrow V;
                 senão
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 34: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
       se x < pt.chave então
            RemoverAVL(x, pt.esq, h)
            balancear(pt, E, h)
       senão
            se x > pt.chave então
                 RemoverAVL(x, pt.dir, h)
                 balancear(pt, D, h)
            senão
10
                  aux \leftarrow pt
                 se pt.esa = \lambda então
                      pt \leftarrow pt.dir; h \leftarrow V;
                 senão
                       se pt.dir = \lambda então
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Arvore Binári de Busca AV

```
Algoritmo 35: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
             RemoverAVL(x, pt.esq, h)
             balancear(pt, E, h)
        senão
             se x > pt.chave então
                  RemoverAVL(x, pt.dir, h)
                  balancear(pt. D. h)
             senão
10
                  aux \leftarrow pt
                  se pt.esa = \lambda então
                       pt \leftarrow pt.dir; h \leftarrow V;
                  senão
                       se pt.dir = \lambda então
                            pt \leftarrow pt.esa; h \leftarrow V;
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
Algoritmo 36: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
             RemoverAVL(x, pt.esq, h)
             balancear(pt, E, h)
        senão
             se x > pt.chave então
                  RemoverAVL(x, pt.dir, h)
                  balancear(pt. D. h)
             senão
                  aux \leftarrow pt
                  se pt.esa = \lambda então
                       pt \leftarrow pt.dir; h \leftarrow V;
                  senão
                       se pt.dir = \lambda então
                            pt \leftarrow pt.esa; h \leftarrow V;
16
                       senão
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 37: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
             RemoverAVL(x, pt.esq, h)
             balancear(pt, E, h)
        senão
             se x > pt.chave então
                  RemoverAVL(x, pt.dir, h)
                  balancear(pt. D. h)
             senão
10
                   aux \leftarrow pt
                  se pt.esa = \lambda então
                       pt \leftarrow pt.dir; h \leftarrow V;
                  senão
                        se pt.dir = \lambda então
                             pt \leftarrow pt.esa; h \leftarrow V;
16
                        senão
                             s \leftarrow pt.dir
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotações

Arvore Binár de Busca AV

```
Algoritmo 38: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
             RemoverAVL(x, pt.esq, h)
             balancear(pt, E, h)
        senão
             se x > pt.chave então
                  RemoverAVL(x, pt.dir, h)
                  balancear(pt. D. h)
             senão
                   aux \leftarrow pt
                  se pt.esa = \lambda então
                       pt \leftarrow pt.dir; h \leftarrow V;
                  senão
                        se pt.dir = \lambda então
                             pt \leftarrow pt.esa; h \leftarrow V;
16
                        senão
                             s \leftarrow pt.dir
                             se s.esq = \lambda então
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 39: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
             RemoverAVL(x, pt.esq, h)
             balancear(pt, E, h)
        senão
             se x > pt.chave então
                   RemoverAVL(x, pt.dir, h)
                   balancear(pt. D. h)
             senão
                   aux \leftarrow pt
                   se pt.esa = \lambda então
                        pt \leftarrow pt.dir; h \leftarrow V;
                   senão
                         se pt.dir = \lambda então
                              pt \leftarrow pt.esa; h \leftarrow V;
16
                        senão
18
                              s \leftarrow pt.dir
                              se s.esq = \lambda então
                                   s.esq \leftarrow pt.esq
20
                                   s.bal \leftarrow pt.bal
                                   pt \leftarrow s
22
                                   .
h ← V
23
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotacões

Árvore Binár de Busca AV

```
Algoritmo 40: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
              RemoverAVL(x, pt.esq, h)
              balancear(pt, E, h)
        senão
              se x > pt.chave então
                   RemoverAVL(x, pt.dir, h)
                   balancear(pt. D. h)
              senão
                   aux \leftarrow pt
                   se pt.esa = \lambda então
                        pt \leftarrow pt.dir; h \leftarrow V;
                   senão
                         se pt.dir = \lambda então
                              pt \leftarrow pt.esa; h \leftarrow V;
16
                        senão
                              s \leftarrow pt.dir
18
                              se s.esq = \lambda então
                                    s.esq \leftarrow pt.esq
20
21
                                    s.bal \leftarrow pt.bal
                                    nt \leftarrow s
22
                                    h \leftarrow V
23
                              senão
```

EDA - Aula 15

Prof.

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 41: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
              RemoverAVL(x, pt.esq, h)
              balancear(pt, E, h)
        senão
              se x > pt.chave então
                   RemoverAVL(x, pt.dir, h)
                   balancear(pt. D. h)
              senão
10
                    aux \leftarrow pt
                   se pt.esa = \lambda então
                         pt \leftarrow pt.dir; h \leftarrow V;
                    senão
                         se pt.dir = \lambda então
                              pt \leftarrow pt.esa; h \leftarrow V;
16
                         senão
                              s \leftarrow pt.dir
18
                              se s.esa = \lambda então
                                    s.esq \leftarrow pt.esq
20
21
                                    s.bal \leftarrow pt.bal
                                    nt \leftarrow s
22
                                    h \leftarrow V
23
24
                              senão
                                    enquanto s.esq \neq \lambda faça
25
                                         PaiS \leftarrow s
27
                                         s \leftarrow s.esa
                                    trocar(pt. PaiS.esg)
                                    RemoverAVL(x, pt.dir, h)
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotacões

Árvore Binári de Busca AV

```
Algoritmo 42: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
        se x < pt.chave então
              RemoverAVL(x, pt.esq, h)
              balancear(pt, E, h)
        senão
              se x > pt.chave então
                   RemoverAVL(x, pt.dir, h)
                   balancear(pt. D. h)
              senão
10
                   aux \leftarrow pt
                   se pt.esa = \lambda então
                        pt \leftarrow pt.dir; h \leftarrow V;
                   senão
                         se pt.dir = \lambda então
                              pt \leftarrow pt.esa; h \leftarrow V;
16
                         senão
                              s \leftarrow pt.dir
18
                              se s.esa = \lambda então
                                    s.esq \leftarrow pt.esq
20
21
                                    s.bal \leftarrow pt.bal
                                    nt \leftarrow s
22
                                    h \leftarrow V
23
24
                              senão
                                    enquanto s.esq \neq \lambda faça
25
                                         PaiS \leftarrow s
27
                                         s \leftarrow s.esa
                                    trocar(pt. PaiS.esg)
                                    RemoverAVL(x, pt.dir, h)
                              balancear(pt. D. h)
30
```

EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binári de Busca AV

```
Algoritmo 43: RemoverAVL(x, pt, h)
1 se pt = \lambda então "elem. não existe"; h \leftarrow F;
2 senão
         se x < pt.chave então
              RemoverAVL(x, pt.esq, h)
              balancear(pt, E, h)
         senão
              se x > pt.chave então
                   RemoverAVL(x, pt.dir, h)
                   balancear(pt. D. h)
              senão
10
                    aux \leftarrow pt
                   se pt.esa = \lambda então
                        pt \leftarrow pt.dir; h \leftarrow V;
                    senão
                         se pt.dir = \lambda então
                              pt \leftarrow pt.esa; h \leftarrow V;
16
                         senão
                              s \leftarrow pt.dir
18
                              se s.esa = \lambda então
                                    s.esq \leftarrow pt.esq
20
21
                                    s.bal \leftarrow pt.bal
                                    nt \leftarrow s
22
                                    h \leftarrow V
23
24
                              senão
                                    enquanto s.esq \neq \lambda faça
25
                                         PaiS ← s
27
                                         s \leftarrow s.esa
                                    trocar(pt. PaiS.esg)
28
                                    RemoverAVL(x, pt.dir, h)
29
                              balancear(pt. D. h)
30
31
                    liberar(aux)
```

Bibliografia

EDA - Aula 15

Prof. Eurinard

Aulas Passada

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). EDA - Aula 15

Prof. Eurinardo

Aulas Passadas

Rotações

Árvore Binária de Busca AVI

Algoritmo Remover

Obrigado!