MAC0323 Algoritmos e Estruturas de Dados II

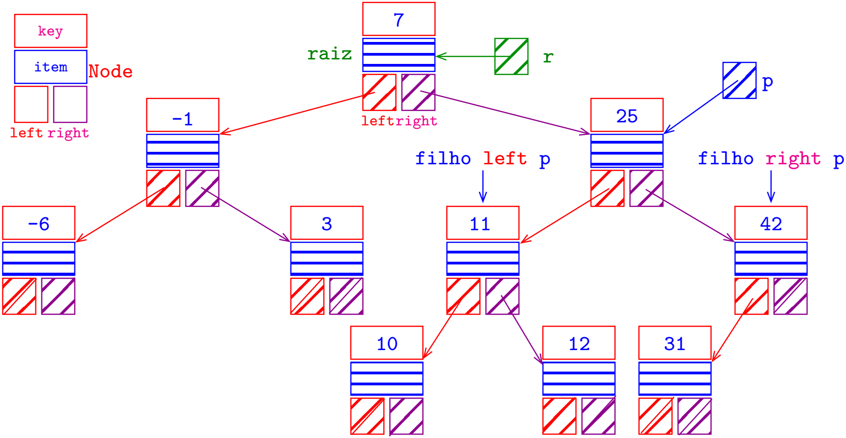
Edição 2020 – 2

Fonte: ash.atozviews.com

Compacto dos melhores momentos

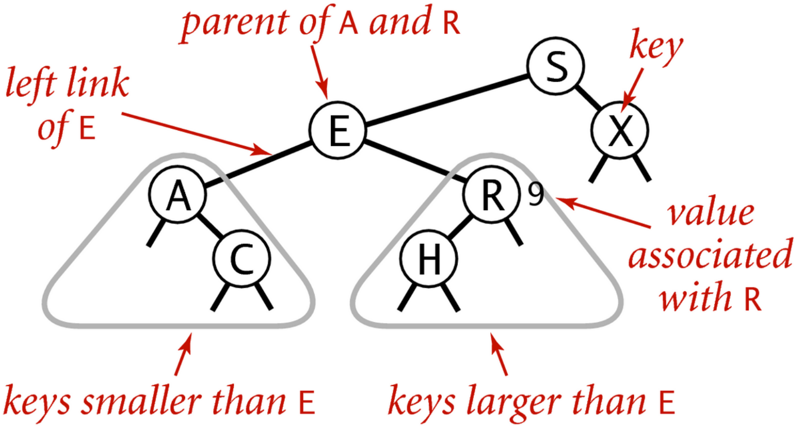
AULA 11

Árvore binárias de busca



in-ordem (e-r-d): −6 −1 3 7 10 11 12 25 31 42

Anatomia de uma árvore binária de busca



Anatomy of a binary search tree

Fonte: `algs4`

Consumo de tempo

O consumo de tempo das funções
`get()`, `put()` e `delete()` é, no pior caso,
proporcional à altura da árvore.

Conclusão: interessa trabalhar com árvores balanceadas: árvores AVL, árvores rubro-negras, árvores . . .

Mais experimentos

Consumo de tempo para se criar uma ST em que a chaves são as palavras em `les_miserables.txt` e os valores o número de ocorrências.

estrutura	ST	tempo
vetor	não-ordenada	59.5
vetor MTF	não-ordenada	7.6
vetor	ordenada	1.5
lista ligada	não-ordenada	147.1
lista ligada MTF	não-ordenada	15.3
lista ligada	ordenada	115.2
skiplist	ordenada	1.1
árvore binária de busca ♡	ordenada	0.7

Tempos em segundos obtidos com StopWatch.

AULA 12

BST aumentada

```
static Node r; /* raiz */
static int n;

struct node {
    Key key;  Value val;
    int n;
    Node *left, *right;
}

Node newNode(Key key, Value val, int n) {
    Node p = mallocSafe(sizeof(*p));
    p->key = key;     p->val = val;
    p->n = n;
    p->left = NULL;   p->right = NULL;
    return p;
}
```

BST: size()

Retorna o número de pares key-val na BST.

```
int size() {
   return sizeTree(r);
}

/* retorna o número de nós na BST de raiz x */
static int sizeTree(Node x) {
   if (x == NULL) return 0;
   return x->n;
}
```

BST: `rank()`

Retorna o número de chaves estritamente menores que `key`.

```
int rank(Key key) {
   return rankTree(key, r);
}
```

BST: rank()

Retorna o número de chaves estritamente menores que key.

```
static int rankTree(Key key, Node x) {
   if (x == NULL) return 0;
   int cmp = compare(key, x->key);
   if (cmp < 0)
      return rankTree(key, x->left);
   if (cmp > 0)
      return 1 + sizeTree(x->left)
               + rankTree(key, x->right);
   return sizeTree(x->left);
}
```

BST: putTree(key, val) aumentado

```
static Node putTree(Node x, Key key, Value val) {
    if (x == NULL)
        return newNode(key, val, 1);
    int cmp = compare(key, x->key);
    if (cmp < 0)
        x->left = putTree(x->left, key, val);
    else if (cmp > 0)
        x->right = putTree(x->right, key, val);
    else
        x->val = val;
    x->n = 1 + sizeTree(x->left) + sizeTree(x->right);
    return x;
}
```

BST: put(key, val) aumentado

```
static Node deleteMinTree(Node x) {
   if (x->left == NULL)
      return x->right;
   x->left = deleteMinTree(x->left);
   x->n = sizeTree(x->left) +
          sizeTree(x->right) + 1;
   return x;
}
```

Desempenho esperado

A ideia de `BST` é realizarmos uma espécie de "*busca binária dinâmica*".

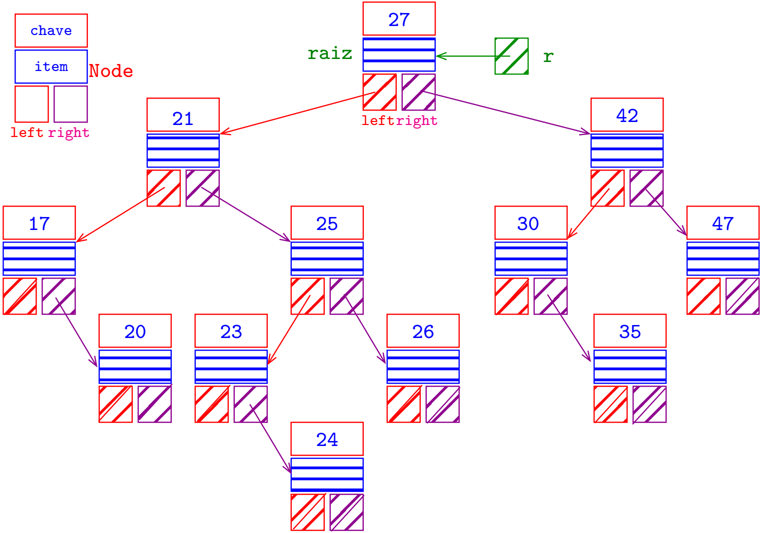
Gostaríamos de combinar a eficiência da busca por chaves ordenadas (`get()`) com a eficiência da inserção (`put()`) e remoção (`delete()`).

Qual é o número de chaves examinadas/comparadas em uma busca com sucesso?

Busca com sucesso = busca em que a chave procurada está na `BST`.

Toda operação de busca ou inserção visita $1 + \mathtt{p}$ nós, sendo `p` a profundidade do último nó visitado.

Desempenho esperado



Desempenho esperado

Na BST acima:

- a busca por 27 requer 1+0 comparações
- a busca por 21 requer 1+1 comparações
- a busca por 42 requer 1+1 comparações
- a busca por 17 requer 1+2 comparações
- a busca por 25 requer 1+2 comparações
- a busca por 30 requer 1+2 comparações
- a busca por 47 requer 1+2 comparações
- a busca por 20 requer 1+3 comparações
- a busca por 23 requer 1+3 comparações
- a busca por 26 requer 1+3 comparações
- a busca por 35 requer 1+3 comparações
- a busca por 24 requer 1+4 comparações

Desempenho esperado

Assim, o número médio de comparações necessárias para uma busca com sucesso na BST acima é

$$(1+2+2+3+3+3+3+4+4+4+4+5)/12 \approx 3.17$$

O **comprimento interno** (= *internal path length*) de uma BT é a soma das profundidades dos seus nós.

O comprimento interno da árvore mostrada anteriormente é

$$0+1+1+2+2+2+2+3+3+3+3+4=26$$

BST aleatória

Uma **BST aleatória** é uma BST que se obtém inserindo `n` chaves distintas em ordem aleatória numa árvore inicialmente vazia.

BST aleatória

Uma **BST aleatória** é uma BST que se obtém inserindo `n` chaves distintas em ordem aleatória numa árvore inicialmente vazia.

Qual é o número esperado de comparações em uma busca com sucesso em uma BST aleatória com `n` chaves?

BST aleatória

Uma **BST aleatória** é uma BST que se obtém inserindo n chaves distintas em ordem aleatória numa árvore inicialmente vazia.

Qual é o número esperado de comparações em uma busca com sucesso em uma BST aleatória com n chaves?

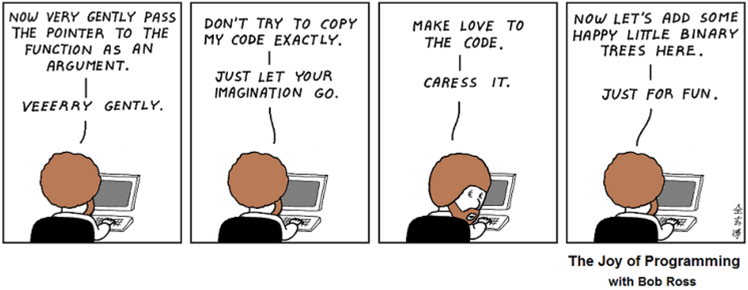
Fato. Buscas com sucesso numa BST aleatória com n chaves requer cerca de $2 \lg n$ comparações na média.

Consumo de tempo

A demonstração do fato anterior
é vista em MAC0338.

O número esperado de nós visitados
durante uma busca em uma BST aleatória
não passa de $2 \lg \mathtt{n}$

Árvores 2-3



Fonte: `https://br.pinterest.com/`

Referências: Árvores 2-3 (PF); Balanced Search Trees (S&W); slides (S&W)

Árvores 2-3

Como implementar uma tabela de símbolos em uma BST de modo que a árvore permaneça aproximadamente balanceada?

Desejamos que a BST tenha altura próxima de $\lg n$, sendo n o número de nós, qualquer que seja a sequência de buscas e inserções aplicada à árvore.

Veremos árvores 2-3 que resolvem o problema em princípio.

A implementação da ideia, usando **árvores rubro-negras**, ainda será discutida.

Árvore 2-3

Uma **árvore 2-3** é:

- uma árvore vazia;
- ou um nó simples com **2 links**:
  - um link `left` para uma árvore 2-3;
  - um link `right` para uma árvore 2-3;
- ou um nó duplo com **3 links**:
  - um link `left` para uma árvore 2-3;
  - um link `mid` para uma árvore 2-3; e
  - um link `right` para uma árvore 2-3.

Árvores 2-3 têm esse nome
porque cada nó tem **2 ou 3 links**.

Ilustração de árvore 2-3

```
                       root----------+
                                     |
                                     V
                                  +-----+
                 +----------------|     |--------------------+
                                  +-----+
                 |                                           |
                 |                                           |
                 V                                           V
            +----+----+                                   +-----+
     +------|    |    |------+                      +-----|     |-----+
     |      +----+----+      |                      |     +-----+     |
     |           |           |                      |                 |
     V           V           V                      V                 V
+----+----+   +-----+     +-----+                +-----+         +----+----+
|    |    |   |     |  +--|     |                |     |--+      |    |    |
+----+----+   +-----+  |  +-----+                +-----+  |      +----+----+
                       |                                  |
                       V                                  V
                    +-----+                          +----+----+
                    |     |                          |    |    |
                    +-----+                          +----+----+
```

Árvore 2-3 perfeitamente balanceadas

Nossas árvores 2-3 são **perfeitamente balanceada**: todos os links `NULL` estão no mesmo nível.

```
                       root----------+
                                     |
                                     V
                                   +----+
                 +-----------------|    |--------------------+
                                   +----+
                 |                                           |
                 |                                           |
                 V                                           V
            +----+----+                                    +----+
     +------|    |    |------+                      +------|    |------+
     |      +----+----+      |                      |      +----+      |
     |           |           |                      |                  |
     V           V           V                      V                  V
+----+----+    +----+      +----+                +----+           +----+----+
|    |    |    |    |      |    |                |    |           |    |    |
+----+----+    +----+      +----+                +----+           +----+----+
```

Estrutura

Importante. Para nós, árvore 2-3 é sinônimo de árvore 2-3 **perfeitamente balanceada**.

Fato. Toda árvore 2-3 de altura $\mathtt{h}$ tem no mínimo $2^{\mathtt{h}+1}-1$ nós e no máximo $3^{\mathtt{h}+1}-1$ nós.

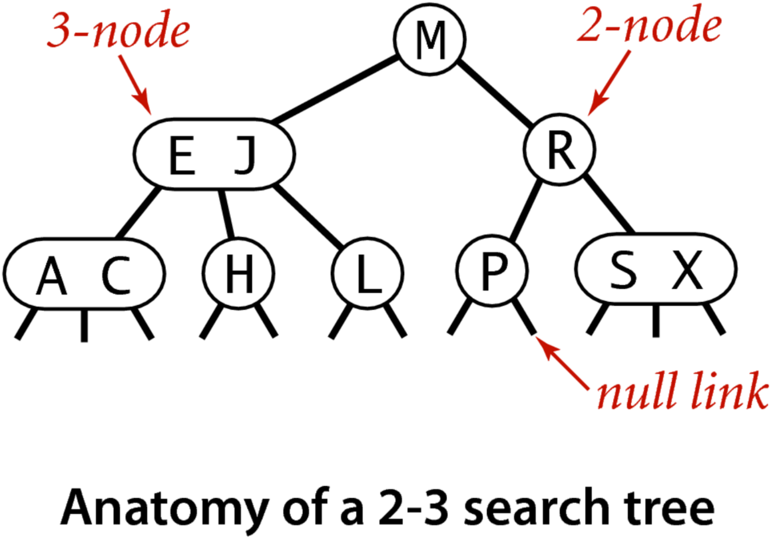
Consequência. Toda árvore 2-3 com $\mathtt{n}$ nós tem altura não superior a $\lg(\mathtt{n}+1)-1$ e não inferior a $\log_3(\mathtt{n}+1)-1$.

Árvore 2-3 de busca

Uma árvore 2-3 de **busca** (*2-3 search tree*) é:

- uma árvore vazia;
- ou um nó simples com uma chave e **2 links**:
  - um link `left` para uma árvore 2-3 que tem chaves menores que a chave do nó e
  - um link `right` para uma árvore 2-3 que tem chaves maiores;
- ou um nó duplo com duas chave e **3 links**:
  - um link `left` para uma árvore 2-3 que tem chaves menores;
  - um link `mid` para uma árvore 2-3 que tem chaves entre as duas chaves do nó; e
  - um link `right` para uma árvore 2-3 que tem chaves maiores.

Anatomia de uma árvore 2-3 de busca



Fonte: algs4

Exemplo de árvore 2-3 de busca

```
                       root----------+
                                     |
                                     V
                                   +----+
                 +-----------------| 45 |--------------------+
                                   +----+
                 |                                           |
                 |                                           |
                 V                                           V
            +----+----+                                    +----+
     +------| 12 | 27 |------+                      +------| 64 |------+
     |      +----+----+      |                      |      +----+      |
     |           |           |                      |                  |
     V           V           V                      V                  V
+----+----+    +----+      +----+                +----+           +----+----+
|  7 | 10 |    | 20 |      | 34 |                | 53 |           | 80 | 95 |
+----+----+    +----+      +----+                +----+           +----+----+
```

Missão

Missão. Manter uma árvore 2-3 de busca sujeita a operações de atualização como `put()`, `deleteMin()`, `delete()`, ...

put(60)

```
                       root----------+
                                     |
                                     V
                                   +----+
                 +-----------------| 45 |--------------------+
                 |                 +----+                    |
                 |                                           |
                 V                                           V
            +----+----+                                 +----+
     +------| 12 | 27 |------+                      +------| 64 |------+
     |      +----+----+      |                      |      +----+      |
     |           |           |                      |                  |
     V           V           V                      V                  V
+----+----+    +----+      +----+                +----+           +----+----+
|  7 | 10 |    | 20 |      | 34 |                | 53 |           | 80 | 95 |
+----+----+    +----+      +----+                +----+           +----+----+
```

put(60)

```
                      root----------+
                             -->    |
                                    V
                                  +-----+       -->
                 +----------------| 45 |--------------------+
                 |                +-----+                   | |
                 |                                          | V
                 V                                          V
            +----+----+                              <--  +-----+
     +------| 12 | 27 |------+                     +------| 64 |------+
     |      +----+----+      |                   | |      +-----+     |
     |           |           |                   V |                  |
     V           V           V                     V                  V
+----+----+    +-----+    +-----+             +----+----+        +----+----+
|  7 | 10 |    | 20 |     | 34 |              | 53 | 60 |        | 80 | 95 |
+----+----+    +-----+    +-----+             +----+----+        +----+----+
```

Inserção em um nó simples (**não** estraga estrutura):
Procura 60 e transforma um 2-nó em 3-nó.

put(25)

```
                        root----------+
                               -->    |
                                      V
                           <--      +----+
                  +-----------------| 45 |-------------------+
               |  |                 +----+                   |
               V  |                                          |
                  V                                          V
             +----+----+                                   +----+
     +-------| 12 | 27 |------+                     +------| 64 |------+
     |       +----+----+      |                     |      +----+      |
     |         |  |           |                     |                  |
     V         V  V           V                     V                  V
+----+----+  +----+----+    +----+             +----+----+        +----+----+
|  7 | 10 |  | 20 | 25 |    | 34 |             | 53 | 60 |        | 80 | 95 |
+----+----+  +----+----+    +----+             +----+----+        +----+----+
              novo 3-nó
```

Procura 25 e transforma um 2-nó em 3-nó.

put(85)

```
                       root----------+
                              -->    |
                                     V
                                   +-----+      -->
                 +-----------------| 45  |------------------+
                 |                 +-----+                  |  |
                 |                                          |  V
                 V                                          V
            +----+----+                                   +-----+ -->
     +------| 12 | 27 |------+                     +------| 64  |-----+
     |      +----+----+      |                     |      +-----+     |  |
     |           |           |                     |                  |  V
     V           V           V                     V                  V
+----+----+ +----+----+    +-----+            +----+----+     +----+----+----+
|  7 | 10 | | 20 | 25 |    | 34  |            | 53 | 60 |     | 80 | 85 | 95 | :-(
+----+----+ +----+----+    +-----+            +----+----+     +----+----+----+
```

Inserção em um nó duplo (**estraga** estrutura): procura 85 e transforma um 3-nó em 4-nó.

put(85)

```
                       root----------+
                                     |
                                     V
                                   +----+
                 +-----------------| 45 |------------------+
                 |                 +----+                  |
                 |                                         |
                 V                                         V
            +----+----+                                  +----+
     +------| 12 | 27 |------+                    +------| 64 |------+
     |      +----+----+      |                    |      +----+      |
     |           |           |                    |                  |
     V           V           V                    V                  V
+----+----+ +----+----+    +----+            +----+----+          +----+
|  7 | 10 | | 20 | 25 |    | 34 |            | 53 | 60 |       +--| 85 |--+ espatifou
+----+----+ +----+----+    +----+            +----+----+       |  +----+  |    o 4-nó
                                                               |          |
                                                               V          V
                                                            +----+      +----+
                                                            | 80 |      | 95 | :-(
                                                            +----+      +----+
                                                                        olha o nível!
```

Espatifa o 4-nó.

put(85)

```
                      root----------+
                                    |
                                    V
                                  +-----+
            +-----------------| 45 |--------------------+
            |                     +-----+                    |
            |                                                |
            V                                                V
       +----+----+                                      +----+----+
  +------| 12 | 27 |------+        :-) novo 3-nó +----| 64 | 85 |----+
  |      +----+----+      |                    |    +----+----+    |
  |           |           |                    |         |         |
  V           V           V                    V         V         V
+----+----+ +----+----+ +-----+          +----+----+ +-----+  +-----+
|  7 | 10 | | 20 | 25 | | 34 |           | 53 | 60 | | 80 |   | 95 |
+----+----+ +----+----+ +-----+          +----+----+ +-----+  +-----+
```

Transforma o 2-nó pai em 3-nó.

put(5)

```
                           root----------+
                                   -->   |
                                         V
                              <--      +----+
                      +----------------| 45 |-------------------+
                    | |                +----+                   |
                    V |                                         |
                      V                                         V
            <--  +----+----+                               +----+----+
        +--------| 12 | 27 |------+                    +---| 64 | 85 |----+
      | |        +----+----+      |                    |   +----+----+    |
      V |             |           |                    |        |         |
        V             V           V                    V        V         V
+----+----+----+ +----+----+   +----+             +----+----+ +----+    +----+
|  5 |  7 | 10 | | 20 | 25 |   | 34 |             | 53 | 60 | | 80 |    | 95 |
+----+----+----+ +----+----+   +----+             +----+----+ +----+    +----+
  novo 4-nó :-(
```

Procura 5 e transforma um 3-nó em 4-nó.

put(5)

```
                                 root------------+
                                                 |
                                                 V
                                               +----+
                           +-------------------| 45 |----------------------+
                           |                   +----+                      |
                           |                                               |
                           V                                               V
                      +----+----+                                     +----+----+
           +----------| 12 | 27 |-------+                         +---| 64 | 85 |----+
           |          +----+----+       |                         |   +----+----+    |
           |               |            |                         |        |         |
           V               V            V                         V        V         V
        +----+        +----+----+     +----+                 +----+----+ +----+     +----+
    +---|  7 |---+    | 20 | 25 |     | 34 |                 | 53 | 60 | | 80 |     | 95 |
    |   +----+   |    +----+----+     +----+                 +----+----+ +----+     +----+
    | espatifou  |
    V   4-nó     V
+----+         +----+
|  5 |         | 10 |     :-(  o nível...
+----+         +----+
```

Espatifa o 4-nó.

put(5)

```
                             root----------+
                                           |
                                           V
                                         +----+
                     +-------------------| 45 |--------------------+
                     |                   +----+                    |
                     |                                             |
         :-( 4-nó    V                                             V
             +----+----+----+                                 +----+----+
   +---------|  7 | 12 | 27 |------+                      +---| 64 | 85 |----+
   |         +----+----+----+      |                      |   +----+----+    |
   |              |    |           |                      |        |         |
   |          +---+    |           |                      |        |         |
   |          |        |           |                      |        |         |
   V          V        V           V                      V        V         V
+----+     +----+ +----+----+    +----+              +----+----+ +----+    +----+
|  5 |     | 10 | | 20 | 25 |    | 34 |              | 53 | 60 | | 80 |    | 95 |
+----+     +----+ +----+----+    +----+              +----+----+ +----+    +----+
```

Transforma o 3-nó pai em 4-nó.

put(5)

```
                             root---+
                                    |
                                    +--->+----+
                     +-------------------| 45 |--------------------+
                     |                   +----+                    |
                     |                                             |
 espatifou o 4-nó    V                                             V
                  +----+                                      +----+----+
              +---| 12 |---+                              +---| 64 | 85 |----+
              |   +----+   |                              |   +----+----+    |
              |            |                              |        |         |
              |            |                              |        |         |
              |            |                              |        |         |
              V            V                              V        V         V
      +----+             +----+                      +----+----+ +----+    +----+
   +--|  7 |--+        +-| 27 |---+                  | 53 | 60 | | 80 |    | 95 |
   |  +----+  |        | +----+   |                  +----+----+ +----+    +----+
   |          |        |          |
   |          |        |          |
   |          |        |          |
   V          V        V          V
+----+     +----+ +----+----+   +----+
|  5 |     | 10 | | 20 | 25 |   | 34 |  :-( nível
+----+     +----+ +----+----+   +----+
```

Espatifa 4-nó.

put(5)

```
                      root----------+
                                    |
                                    V      :-) novo 3-nó
                               +----+----+
          +--------------------| 12 | 45 |--------------------+
          |                    +----+----+                    |
          |                         |                         |
          |                         |                         |
          V                         V                         V
       +----+                     +----+                 +----+----+
   +---|  7 |---+              +--| 27 |--+          +---| 64 | 85 |----+
   |   +----+   |              |  +----+  |          |   +----+----+    |
   |            |              |          |          |        |         |
   |            |              |          |          |        |         |
   |            |              |          |          |        |         |
   V            V              V          V          V        V         V
+----+        +----+      +----+----+   +----+  +----+----+ +----+    +----+
|  5 |        | 10 |      | 20 | 25 |   | 34 |  | 53 | 60 | | 80 |    | 95 |
+----+        +----+      +----+----+   +----+  +----+----+ +----+    +----+
```

Transforma o 2-nó pai em 3-nó.

put(57)

```
                      root----------+
                             -->    |
                                    V
                               +----+----+       -->
          +--------------------| 12 | 45 |--------------------+
          |                    +----+----+                    |
          |                         |                      |  |
          |                         |                      V  |
          V                         V                         V
       +-----+                    +-----+                +----+----+
   +---|  7  |---+             +--|  27 |--+         +---| 64 | 85 |----+
   |   +-----+   |             |  +-----+  |       | |   +----+----+    |
   |             |             |           |       V |        |         |
   |             |             |           |         +--+     +-----+   +-----+
   |             |             |           |            |           |         |
   V             V             V           V            V           V         V
+-----+       +-----+     +----+----+   +-----+ +----+----+----+ +-----+   +-----+
|  5  |       | 10  |     | 20 | 25 |   | 34  | | 53 | 57 | 60 | | 80  |   | 95  |
+-----+       +-----+     +----+----+   +-----+ +----+----+----+ +-----+   +-----+
                                                 :-( novo 4-nó
```

Procura 57 e transforma o 3-nó em 4-nó.

put(57)

```
                     root----------+
                                   |
                                   V
                              +----+----+
          +-------------------| 12 | 45 |-------------------+
          |                   +----+----+                   |
          |                        |                        |
          |                        |                        |
          V                        V                        V
       +----+                    +----+                +----+----+
   +---|  7 |---+             +--| 27 |--+         +---| 64 | 85 |----+
   |   +----+   |             |  +----+  |         |   +----+----+    |
   |            |             |          |         |        |         |
   |            |             |          |         +---+    +----+   +----+
   |            |             |          |            |          |        |
   V            V             V          V            V          V        V
+----+        +----+     +----+----+   +----+      +----+      +----+   +----+
|  5 |        | 10 |     | 20 | 25 |   | 34 |  +---| 57 |---+  | 80 |   | 95 |
+----+        +----+     +----+----+   +----+  |   +----+   |  +----+   +----+
                                               |            |
                                               |            |
                                               V            V
                                            +----+        +----+
                                            | 53 |        | 60 |  :-( nível
                                            +----+        +----+
```

Espatifa o 4-nó.

put(57)

```
                      root----------+
                                    |
                                    V
                               +----+----+
          +--------------------| 12 | 45 |--------------------+
          |                    +----+----+                    |
          |                         |                         |
          |                         |                         |  :-( 4-nó
          V                         V                         V
       +----+                     +----+               +----+----+----+
   +---|  7 |---+              +--| 27 |--+         +--| 57 | 64 | 85 |-----+
   |   +----+   |              |  +----+  |         |  +----+----+----+     |
   |            |              |          |         |       |    |          |
   |            |              |          |         |       |    +--+       |
   |            |              |          |         |       |       |       |
   V            V              V          V         V       V       V       V
+----+        +----+      +----+----+   +----+   +----+  +----+  +----+  +----+
|  5 |        | 10 |      | 20 | 25 |   | 34 |   | 53 |  | 60 |  | 80 |  | 95 |
+----+        +----+      +----+----+   +----+   +----+  +----+  +----+  +----+
```

Transforma o 3-nó pai em 4-nó.

put(57)

```
                      root----------+
                                    |
                                    V
                               +----+----+
          +--------------------| 12 | 45 |--------------------+
          |                    +----+----+                    |
          |                         |                         |
          |                         |                         |
          V                         V                         V
       +----+                     +----+                    +----+
   +---|  7 |---+              +--| 27 |--+           +-----| 64 |-----+
   |   +----+   |              |  +----+  |           |     +----+     |
   |            |              |          |           |                |
   |            |              |          |           |                |
   V            V              V          V           V                V
+----+        +----+      +----+----+   +----+     +----+            +----+
|  5 |        | 10 |      | 20 | 25 |   | 34 |  +--| 57 |--+      +--| 85 |--+
+----+        +----+      +----+----+   +----+  |  +----+  |      |  +----+  |
                                                |          |      |          |
                                                |          |      |          |
                                                V          V      V          V
                                             +----+      +----+ +----+    +----+
                                :-( nível    | 53 |      | 60 | | 80 |    | 95 |
                                             +----+      +----+ +----+    +----+
```

Espatifa o 4-nó.

put(57)

```
                      root------------+
                                      |
                                      V
                               +----+----+----+
          +--------------------| 12 | 45 | 64 |----------------------+
                               +----+----+----+
          |                         |    |                           |
          |                     +---+    +---------+                 |
          |                     |                  |                 |
          V                     V                  |                 |
       +----+                 +----+               V                 V
   +---|  7 |---+          +--| 27 |--+          +----+            +----+
   |   +----+   |          |  +----+  |       +--| 57 |--+      +--| 85 |--+
   |            |          |          |       |  +----+  |      |  +----+  |
   |            |          |          |       |          |      |          |
   V            V          V          V       V          V      V          V
+----+        +----+  +----+----+   +----+ +----+      +----+ +----+    +----+
|  5 |        | 10 |  | 20 | 25 |   | 34 | | 53 |      | 60 | | 80 |    | 95 |
+----+        +----+  +----+----+   +----+ +----+      +----+ +----     +----+
```

Transforma o 3-nó pai em um 4-nó.

put(57)

```
                       root---+
                              +----->+-----+
                      +--------------| 45  |------------------+
                      |                                       |
                      |                                       |
                      V                                       V
                   +-----+                                 +-----+
          +--------| 12  |--------+                   +----| 64  |--------+
          |        +-----+        |                   |    +-----+        |
          |                       |                   |                   |
          V                       V                   V                   V
       +-----+                 +-----+             +-----+            +-----+
   +---|  7  |---+          +--| 27  |--+       +--| 57  |--+       +--| 85  |--+
   |   +-----+   |          |  +-----+  |       |  +-----+  |       |  +-----+  |
   |             |          |           |       |           |       |           |
   |             |          |           |       |           |       |           |
   V             V          V           V       V           V       V           V
+-----+       +-----+ +-----+-----+  +-----+ +-----+     +-----+ +-----+     +-----+
|  5  |       | 10  | | 20  | 25  |  | 34  | | 53  |     | 60  | | 80  |     | 95  |
+-----+       +-----+ +-----+-----+  +-----+ +-----+     +-----+ +----       +-----+
```

Espatifa o 4-nó. **Altura foi incrementada!**

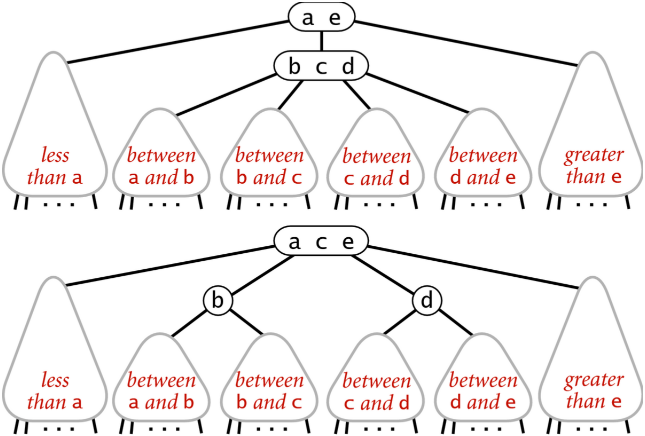
Transformações preservam propriedades

```
                   root------------+
                                   |
                                   V
                            +----+----+----+
       +--------------------| 12 | 45 | 64 |----------------------+
       |                    +----+----+----+                      |
       |                         |    |                           |
       |                +--------+    +---------+                 |
       |                |                       |                 |
       V                V                       V                 V
+--------------+   +------------------+  +------------------+  +-------------+
| chaves < 12  |   | 12 < chaves < 45 |  | 45 < chaves < 64 |  | 64 < chaves |
|              |   |                  |  |                  |  |             |
+--------------+   +------------------+  +------------------+  +-------------+
```

Transformações preservam propriedades

```
          root------------+
                          |
                          V
                        +----+
        +---------------| 45 |-----------------+
        |               +----+                 |
        |                                      |
        V                                      V
      +----+                                 +----+
+-------| 12 |---------+              +------| 64 |------+
|       +----+         |              |      +----+      |
|                      |              |                  |
|                      |              |                  |
V                      V              V                  V
+--------------+  +-----------------+  +-----------------+  +-------------+
| chaves < 12  |  | 12 < chaves < 45 |  | 45 < chaves < 64 |  | 64 < chaves |
|              |  |                  |  |                  |  |             |
+--------------+  +-----------------+  +-----------------+  +-------------+
```

Transformações preservam propriedades



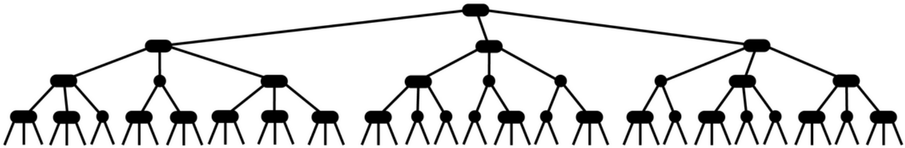
Splitting a 4-node is a local transformation that preserves order and perfect balance

Fonte: algs4

Consumo de tempo

Numa árvore 2-3 com $\mathtt{n}$ nós, busca e inserção nunca visitam mais que $\lg(\mathtt{n}+1)$. Cada visita faz no máximo 2 comparações de chaves.

Árvore 2-3 aleatória



Typical 2-3 tree built from random keys

Fonte: algs4

Implementação

Usaremos `BST`s (árvores binária de busca) para simular árvores 2-3.

BSTs rubro-negras

Fonte: http://scottlobdell.me/

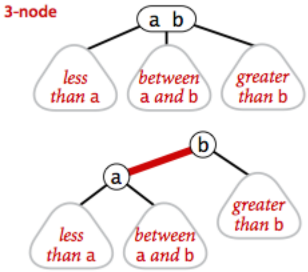
Referências: BSTs rubro-negras (PF); Balanced Search Trees (S&W); slides (S&W) Hashing Functions (S&W)

BSTs rubro-negras

Uma BST rubro-negra (*red-black BST*) é uma BST que simula uma árvores 2-3.

Cada 3-nó da árvore 2-3 é representado por dois 2-nós ligados por um link rubro.

Nossas BSTs são esquerdistas (*left-leaning*), pois os links rubros são **sempre** inclinados para a esquerda.



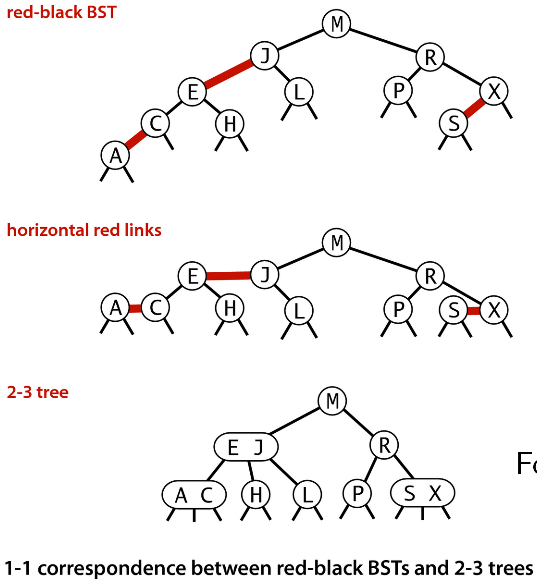
BSTs rubro-negras

Uma **BST rubro-negra** é
uma BST cujos links são negros e rubros e:

- links rubros se inclinam para a esquerda;
- nenhum nó incide em dois links rubros;
- **balanceamento negro perfeito**: todo caminho da raiz até um link NULL tem o mesmo número de links negros.

Se os links rubros forem desenhados horizontalmente e depois contraídos, teremos uma árvore 2-3.

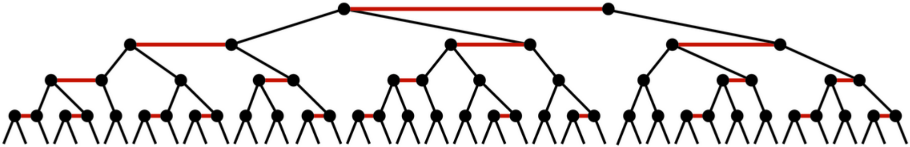
Anatomia de uma árvore rubro-negra



Fonte: algs4

Árvore 2-3 para rubro-negra

Se os links rubros forem desenhados horizontalmente e depois contraídos, teremos uma árvore 2-3:



A red-black tree with horizontal red links is a 2-3 tree

Fonte: algs4

Árvore 2-3 para rubro-negra

```
                       +----+----+
       +---------------| 45 | 64 |--------------+
       |               +----+----+              |
       |                    |                   |
       |                    |                   |
       |                    |                   |
       V                    V                   V
+--------------+   +------------------+  +-------------+
| chaves < 45  |   | 45 < chaves < 64 |  | 64 < chaves |
|              |   |                  |  |             |
+--------------+   +------------------+  +-------------+
```

Árvore 2-3 para rubro-negra

```
                               +-----+
link rubro    --->  +==========| 64 |---------------+  <--- link negro
                    !          +-----+              |
                    !                               |
                    V                               V
                 +-----+                   +-------------+
         +-------| 45 |-------+            | 64 < chaves |
         |       +-----+      |            |             |
         |                    |            +-------------+
         |                    |
         |                    |
         V                    V
  +--------------+   +------------------+
  | chaves < 45  |   | 45 < chaves < 64 |
  |              |   |                  |
  +--------------+   +------------------+

      -------+                              +========
             |  link negro                  !  link rubro
             |                              !
             +------->              <=======+
```

Árvore 2-3

```
                       root----------+
                                     |
                                     V
                                   +----+
                 +-----------------| 45 |--------------------+
                 |                 +----+                    |
                 |                                           |
                 V                                           V
            +----+----+                                 +----+----+
     +------| 12 | 27 |------+                      +---| 64 | 85 |----+
     |      +----+----+      |                      |   +----+----+    |
     |           |           |                      |        |         |
     V           V           V                      V        V         V
+----+----+ +----+----+    +----+              +----+----+ +----+    +----+
|  7 | 10 | | 20 | 25 |    | 34 |              | 53 | 60 | | 80 |    | 95 |
+----+----+ +----+----+    +----+              +----+----+ +----+    +----+
```

Árvore rubro-negra

```
                                      negra
                           root -------------> +-----+
                                +--------------|  45 |-------------------+
                                |              +-----+                   |
                                |                                        |
                                V   negro                                V   negro
                             +-----+                                  +-----+
                      +======|  27 |-----+                     +======|  85 |-------+
                      !      +-----+     |                     !      +-----+       |
                      !                  |                     !                    |
                      V  rubro           V   negro             V   rubro            V
                   +-----+             +-----+              +-----+             +-----+
           +-------|  12 |-----+       |  34 |       +------|  64 |------+      |  95 |
           |       +-----+     |       +-----+       |      +-----+      |      +-----+
           |                   |                     |                   |
           V  negro            V    negro            V   negro           V   negro
       +-----+               +-----+              +-----+             +-----+
   +===|  10 |           +===|  25 |          +===|  60 |             |  80 |
   !   +-----+           !   +-----+          !   +-----+             +-----+
   !                     !                    !
   V   rubro             V   rubro            V   rubro
+-----+               +-----+              +-----+
|  7  |               |  20 |              |  53 |
+-----+               +-----+              +-----+
```

Balanço e profundidade

O **balanço negro perfeito** vem do fato de que os links negros correspondem aos links da árvore 2-3.

Nota. No CLRS, as árvores rubro-negras têm nós rubros e negros:

- nós rubros são os referenciados por links rubros.
- **nós negros** são os referenciados por **links negros**.

A **profundidade negra** de um nó `x` é o número de **links negros** no caminho da raiz até `x`.

A **altura negra** da árvore é o máximo da **profundidade negra** de todos os nós.

Nós de uma BST rubro-negra

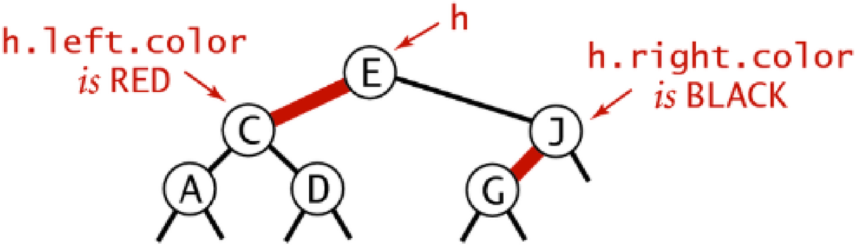
É inconveniente armazenar a cor de um link na estrutura de dados; é mais simples armazenar essa informação nos nós.

A cor de um nó é a cor do único link que entra nele.

A raiz é considerada **negra**.

```
static bool RED = true;
static bool BLACK = false;
```

Nós de uma BST rubro-negra



Nós de uma BST rubro-negra

```
typedef struct node *Node;

struct node {
    Key key;    Value val;
    int n; /* número de nós nesta subárvore */
    bool color; /* cor do link para este nó */
    Node left, right;
}

Node newNode(Key key, Value val,
                int n, bool color) {
    Node p = mallocSafe(sizeof(*p));
    p->key = key;      p->val = val;
    p->n = n;          p->color = color;
    p->left = NULL;    p->right = NULL;
    return p;
}
```

Nós de uma BST rubro-negra

```
typedef struct node *Node;

struct node {
   Key key;    Value val;
   int n; /* número de nós nesta subárvore */
   bool color; /* cor do link para este nó */
   Node left, right;
}

static bool isRed(Node x) {
   if (x == NULL) return false;
   return x->color == RED;
}
```

get(key)

O código de busca (= get()) para BSTs rubro-**negras** é exatamente igual ao das BSTs comuns!

```
Value get(Key key) {
   Node x = getTree(r, key);
   if (x == NULL) return NULL;
   return x->val;
}
```

get(key)

O código de busca (= `get()`) para BSTs **rubro-negras** é exatamente igual ao das BSTs comuns!

```
static Node getTree(Node x, Key key) {
   /* Considera subárvore que tem raiz x */
   if (x == NULL) return NULL;
   int cmp = compare(key, x->key);
   if (cmp < 0)
      return getTree(x->left, key);
   else if (cmp > 0)
      return getTree(x->right, key);
   else return x;
}
```

get(key) versão iterativa

Recebe uma chave key e retorna o valor val associado a key; se key não está na BST, retorna NULL.

```
static Node getTree(Node x, Key key) {
    if (x == NULL) return NULL;
    while (x != NULL && compare(key, x->key) != 0)
        int cmp = compare(key, x->key);
        if (cmp > k)
            x = x->left;
        else
            x = x->right;
    return x;
}
```

Inserção em um 2-nó

Árvore formada por apenas um 2-nó

```
root -------+                    root -------+
            |                                |
            V   negro                        V
          +----+                           +----+
          | 53 |                           | 53 | 2-nó
          +----+                           +----+
```

Inserção em um 2-nó

Árvore formada por apenas um 2-nó

```
put(50)

root -------+                  root -----------+
            |                                  |
            V   negro                          V
          +-----+                        +-----+-----+
     +====| 53 |                         | 50 | 53 |
     !    +-----+                        +-----+-----+
     !
     !
     V   rubro
  +-----+
  | 50 |
  +-----+
```

Inserção em um 2-nó

Árvore formada por apenas um 2-nó

```
put(60)

root --------+                root -----------+
             |                                |
             V   negro                        V
           +-----+                       +-----+-----+
           | 53  |===+                   | 53  | 60  |
           +-----+   !                   +-----+-----+
                     !
     +-------+       !
     |       |       V   rubro
     | left  |    +-----+
     +->     |    | 60  |
                  +-----+
```

Inserção em um 2-nó

Árvore formada por apenas um 2-nó

```
root = rotateLeft(root);

root -------+                root -----------+
            |                                |
            V   negro                        V
          +----+                          +----+----+
     +====| 60 |                          | 53 | 60 |
     !    +----+                          +----+----+
     !
     !
     V   rubro
   +----+
   | 53 |
   +----+
```

Inserção em um 2-nó qualquer

```
root --------+
             |
             V   negro
           +----+
    +------| 53 |-----+
    |      +----+     |
    |                 |
    V  negro          V    negro
 +----+             +----+
 | 40 |         +===| 74 |
 +----+         !   +----+
                !
                V   rubro
             +----+
             | 62 |
             +----+
```

```
root ----------+
               |
               V
             +----+
      +------| 53 |------+
      |      +----+      |
      |                  |
      V                  V
   +----+           +----+----+
   | 40 |           | 62 | 74 |
   +----+           +----+----+
    2-nó
```

Inserção em um 2-nó qualquer

```
put(32)
      root --------+                    root ----------+
                   |                                   |
                   V   negro                           V
                 +----+                              +----+
          +------| 53 |-----+                 +------| 53 |------+
          |      +----+     |                 |      +----+      |
          |                 |                 |                  |
          V   negro         V    negro        V                  V
       +----+             +----+         +----+----+        +----+----+
   +===| 40 |         +===| 74 |         | 32 | 40 |        | 62 | 74 |
   !   +----+         !   +----+         +----+----+        +----+----+
   !                  !
   V   rubro          V   rubro
+----+             +----+
| 32 |             | 62 |
+----+             +----+
```

Inserção em um 2-nó qualquer

put(49)

```
       root ----------+                    root ----------+
                      |                                   |
                      V   negro                           V
                    +----+                              +----+
            +-------| 53 |-----+                 +------| 53 |------+
            |       +----+     |                 |      +----+      |
h           |                  |                 |                  |
|           V   negro          V    negro        V                  V
+------->+----+              +----+         +----+----+        +----+----+
         | 40 |===+      +===| 74 |         | 40 | 49 |        | 62 | 74 |
         +----+   !      !   +----+         +----+----+        +----+----+
                  !      !
      +------+    V      V   rubro
      | left | +----+ +----+
      |      | | 49 | | 62 |
      +-->   | +----+ +----+
                rubro
```

Inserção em um 2-nó qualquer

```
h = rotateLeft(h);

       root ----------+                    root ----------+
                      |                                   |
                      V   negro                           V
                    +----+                              +----+
            +-------| 53 |-----+                 +------| 53 |------+
            |       +----+     |                 |      +----+      |
 h          |                  |                 |                  |
 |          V   negro          V    negro        V                  V
 +------>+----+              +----+         +----+----+        +----+----+
     +===| 49 |          +===| 74 |         | 40 | 49 |        | 62 | 74 |
     !   +----+          !   +----+         +----+----+        +----+----+
     !                   !
     V   rubro           V   rubro
  +----+              +----+
  | 40 |              | 62 |
  +----+              +----+
```

Inserção em um 3-nó

```
root ----------+
               |
               V   negro
             +-----+
        +====| 53  |
        !    +-----+
        !
        !
        V   rubro
      +-----+
      | 40  |
      +-----+
```

```
root -------------+
                  |
                  V
             +----+----+
             | 40 | 53 |
             +----+----+
                3-nó
```

chave inserida é a menor do 3-nó

```
    put(31)
          root ----------+          root -------------+
                         |                            |
                         V   negro                    V
          h------------->+----+                   +----+----+----+
                  +====| 53 |                   | 31 | 40 | 53 |
                  !    +----+                   +----+----+----+
                  !                                   4-nó
                  !          +-------+
                  V   rubro  |       |
     x -------->+----+       | right |
           +====| 40 |       |       |
           !    +----+          <---+
           !
           !
   rubro   V
        +----+
        | 31 |
        +----+
```

chave inserida é a menor do 3-nó

```
x = rotateRight(x);

      root ----------+                    root -------------+
                     |                                      |
                     V   negro                              V
    h-----------> +-----+                              +----+----+----+
              +====| 40 |====+                         | 31 | 40 | 53 |
              !    +-----+    !                        +----+----+----+
x             !               !                             4-nó
|             !               !
|             V   rubro       V   rubro
+----------->+-----+       +-----+
             | 31 |        | 53 |
             +-----+       +-----+
```

chave inserida é a menor do 3-nó

```
flipColors(h);

  root ==========+                 root -------------+
                 !                                   |
                 V   rubro                           V   2-nó
h------------> +-----+                            +-----+
          +----| 40 |----+                    +---| 40 |---+
          |    +-----+    |                   |   +-----+   |
          |               |                   |             |
          |               |                   |             |
          V   negro       V   negro           V   2-nó      V    2-nó
       +-----+         +-----+             +-----+       +-----+
       | 31 |          | 53 |              | 31 |        | 53 |
       +-----+         +-----+             +-----+       +-----+
```

chave é inserida entre as chaves do 3-nó

```
put(45)
 root ----------+                root --------------+
                |                                   |
                V   negro                           V
 h------------>+-----+                   +----+----+-----+
         +====| 53 |                     | 40 | 45 | 53 |
         !    +-----+                    +----+----+-----+
x        !                                    4-nó
|        !
|        V   rubro
+----->+-----+
       | 40 |====+
       +-----+   !
                 !
    +-------+    !
    | left |     V    rubro
    |       |  +-----+
    +--->   |  | 45 |
               +-----+
```

chave é inserida entre as chaves do 3-nó

```
x = rotateLeft(x);
      root ----------+         root -------------+
                     |                           |
                     V   negro                   V
    h------------> +----+                   +----+----+----+
              +====| 53 |                   | 40 | 45 | 53 |
              !    +----+                   +----+----+----+
x             !                                   4-nó
|             !         +-------+
|             V   rubro |       |
+---------->+----+      | right |
       +====| 45 |      |       |
       !    +----+      +   <---+
       !
       !
       V
    +----+ rubro
    | 40 |
    +----+
```

chave é inserida entre as chaves do 3-nó

```
h = rotateRight(h);

      root ----------+                 root -------------+
                     |                                   |
                     V   negro                           V
    h------------> +----+                        +----+----+----+
              +====| 45 |====+                   | 40 | 45 | 53 |
              !    +----+    !                   +----+----+----+
x             !              !                         4-nó
|             !              !
|             V   rubro      V   rubro
+----------->+----+        +----+
            | 40 |        | 53 |
            +----+        +----+
```

chave é inserida entre as chaves do 3-nó

```
flipColors(h); hmmm. raiz deve ser negra

   root ==========+           root --------------+
                  !                              |
                  V   rubro                      V   2-nó
 h------------> +-----+                       +-----+
           +----| 45 |----+                +---| 45 |---+
           |    +-----+   |                |   +-----+  |
           |              |                |            |
           |              |                |            |
           V   negro      V   negro        V   2-nó     V     2-nó
         +-----+       +-----+          +-----+      +-----+
         | 40 |        | 53 |           | 40 |       | 53 |
         +-----+       +-----+          +-----+      +-----+
```

chave inserida é maior que todas do 3-nó

```
put(77)

root ----------+                root -------------+
               |                                  |
               V   negro                          V
            +-----+                   +----+----+----+
      +====| 53 |====+                | 40 | 53 | 77 |
      !     +-----+     !                +----+----+----+
      !                 !                      4-nó
      !                 !
      V   rubro         V   rubro
   +-----+           +-----+
   | 40 |           | 77 |
   +-----+           +-----+
```

chave inserida é maior que todas do 3-nó

```
flipColors(root);  hmmm. raiz deve ser negra

root ==========+              root -------------+
               !                                |
               V   rubro                        V   2-nó
             +-----+                          +-----+
        +----| 53 |----+                  +---| 53 |---+
        |    +-----+    |                  |   +-----+   |
        |               |                  |             |
        |               |                  |             |
        V   negro       V    negro         V   2-nó      V     2-nó
      +-----+         +-----+            +-----+       +-----+
      | 40 |          | 77 |             | 40 |        | 77 |
      +-----+         +-----+            +-----+       +-----+
```

chave é inserida em um 3-nó qualquer

```
                     root ------+
                                |
                                V   negro
                             +-----+
                      +======| 27  |-----+
                      !      +-----+     |
                      !                  |
                      V  rubro           V   negro
                   +-----+             +-----+
           +-------| 12  |-----+       | 34  |
           |       +-----+     |       +-----+
           |                   |
           V  negro            V    negro
       +-----+               +-----+
   +===| 10  |           +===| 25  |
   !   +-----+           !   +-----+
   !                     !
   V   rubro             V   rubro
+-----+               +-----+
|  7  |               | 20  |
+-----+               +-----+
```

```
         root ---+
                 |
                 |
                 V      3-nó
            +----+----+
     +------| 12 | 27 |------+
     |      +----+----+      |
     |           |           |
     V           V           V
+----+----+ +----+----+    +-----+
|  7 | 10 | | 20 | 25 |    | 34  |
+----+----+ +----+----+    +-----+
  3-nó         3-nó          2-nó
```

chave é inserida em um 3-nó qualquer

```
put(15)
                     root ------+                                root ---+
                                |                                        |
                                V   negro                                |
                             +----+                                      V      3-nó
                      +======| 27 |------+                          +----+----+
                      !      +----+      |                  +-------| 12 | 27 |----------+
                      !                  |                  |       +----+----+          |
                      V  rubro           V   negro          |            |               |
                   +----+              +----+               |            +-+             |
           +-------| 12 |------+       | 34 |               |              |             |
           |       +----+      |       +----+               V              V             V
           |                   |                       +----+----+  +----+----+----+    +----+
           V  negro            V    negro              |  7 | 10 |  | 15 | 20 | 25 |    | 34 |
       +----+                +----+<-------- h         +----+----+  +----+----+----+    +----+
   +===| 10 |            +===| 25 |                      3-nó            4-nó           2-nó
   !   +----+            !   +----+  +-------+
   !                     !           | right |
   V   rubro             V   rubro      <----+
+----+                +----+
|  7 |            +===| 20 |
+----+            !   +----+
                  !
                  V    rubro
                +----+
                | 15 |
                +----+
```

chave é inserida em um 3-nó qualquer

```
h = rotateRight(h);
                     root ------+                               root ---+
                                |                                       |
                                V   negro                               |
                             +-----+                                    V      3-nó
                      +=======| 27 |------+                        +----+----+
                      !      +-----+      |                 +------| 12 | 27 |----------+
                      !                   |                 |      +----+----+          |
                      V  rubro            V   negro         |           |               |
                   +-----+             +-----+              |           +-+             |
           +-------| 12  |------+      | 34  |              |             |             |
           |       +-----+      |      +-----+              V             V             V
           |                    |                      +----+----+ +----+----+----+    +-----+
           V  negro             V   negro              |  7 | 10 | | 15 | 20 | 25 |    | 34  |
       +-----+               +-----+<-------- h        +----+----+ +----+----+----+    +-----+
   +===| 10  |           +===| 20  |===+                 3-nó            4-nó           2-nó
   !   +-----+           !   +-----+   !
   !                     !             !
   V   rubro             V  rubro      V  rubro
+-----+               +-----+       +-----+
|  7  |               | 15  |       | 25  |
+-----+               +-----+       +-----+
```

chave é inserida em um 3-nó qualquer

```
flipColors(h);
                     root ------+                               root ---+
                                |                                       |
                                V   negro                               |
                             +-----+                                    V      4-nó
                      +======| 27 |------+                      +----+----+----+
           h          !      +-----+     |                  +---| 12 | 20 | 27 |----+
           |          !                  |                  |   +----+----+----+    |
           |          V  rubro           V   negro          |        |    |         |
           +------->+-----+           +-----+               |        |    +---+     |
           +-------| 12 |======+       | 34 |               |        |       |      |
           |       +-----+     !       +-----+              V        V       V      V
           |         +-------+ !                       +----+----+ +-----+ +-----+ +-----+
           V  negro  | left |  V    rubro              |  7 | 10 | | 15 |  | 25 | | 34 |
       +-----+       +--->   +-----+                   +----+----+ +-----+ +-----+ +-----+
   +===| 10 |            +---| 20 |---+                  3-nó       2-nó    2-nó    2-nó
   !   +-----+           |   +-----+  |
   !                     |            |
   V   rubro             V  negro     V   negro
+-----+               +-----+       +-----+
|  7 |                | 15 |        | 25 |
+-----+               +-----+       +-----+
```

chave é inserida em um 3-nó qualquer

```
h = rotateLeft(h);
                            root ------+
                                       |
                                       V   negro
                  +---------------->+----+
                  |          +======| 27 |------+
                  h          !      +----+      |
                             !        +-----+   |
                             V rubro  |right|   V   negro
                          +----+            | +----+
                  +=======| 20 |------+  <--+ | 34 |
                  !       +----+      !       +----+
                  !                   !
                  V  rubro            V    negro
              +----+                +----+
          +---| 12 |---+            | 25 |
          |   +----+   |            +----+
          |            |
          V   negro    V
       +----+        +----+
   +===| 10 |        | 15 |
   !   +----+        +----+

    !
   V   rubro
+----+
|  7 |
+----+
```

```
         root ---+
                 |
                 |
                 V      4-nó
         +----+----+----+
     +---| 12 | 20 | 27 |----+
     |   +----+----+----+    |
     |        |    |         |
     |        |    +--+      |
     |        |       |      |
     V        V       V      V
+----+----+ +----+  +----+ +----+
|  7 | 10 | | 15 |  | 25 | | 34 |
+----+----+ +----+  +----+ +----+
  3-nó       2-nó    2-nó   2-nó
```

chave é inserida em um 3-nó qualquer

```
h = rotateRight(h);

                     root ------+                             root ---+
                                |                                     |
                                V   negro                             |
           +---------------->+----+                                   V     4-nó
           |          +======| 20 |=========+                  +----+----+----+
           h          !      +----+         !              +---| 12 | 20 | 27 |----+
                      !                     !              |   +----+----+----+    |
                      V rubro               V   rubro      |        |    |         |
                   +----+                +----+            |        |    +--+      |
           +-------| 12 |------+      +---| 27 |---+       |        |       |      |
           |       +----+      !      |   +----+   |       V        V       V      V
           |                   !      |            |  +----+----+ +----+  +----+ +----+
           V  negro     negro  V      V            V  |  7 | 10 | | 15 |  | 25 | | 34 |
       +----+                +----+ +----+     +----+ +----+----+ +----+  +----+ +----+
   +===| 10 |                | 15 | | 25 |     | 34 |  3-nó       2-nó    2-nó    2-nó
   !   +----+                +----+ +----+     +----+

    !
   V   rubro
+----+
|  7 |
+----+
```

chave é inserida em um 3-nó qualquer

```
flipColors(h);
                  root ======+                                root ----+
                             !                                         |
                             V   rubro                                 V
        +-------------------->+-----+                                +-----+
        |          +---------| 20 |------+                       +---| 20 |----+
        h          |          +-----+     |                       |    +-----+    |
                   |                      |                       |               |
                   V negro                V   negro               V               V
                +-----+                +-----+                 +-----+         +-----+
        +-------| 12 |------+       +---| 27 |---+           +--| 12 |-+      +--| 27 |--+
        |        +-----+     !       |   +-----+   |          |  +-----+ |     |  +-----+  |
        |                    !       |             |          |          |     |           |
        V  negro      negro  V       V negro negroV           V          V     V           V
     +-----+               +-----+ +-----+    +-----+ +----+----+ +-----+ +-----+    +-----+
 +===| 10 |                | 15 | | 25 |     | 34 | |  7 | 10 | | 15 | | 25 |    | 34 |
 !   +-----+               +-----+ +-----+    +-----+ +----+----+ +-----+ +-----+    +-----+
 !
 V   rubro
+-----+
|  7 |
+-----+
root.color = BLACK; /* manter BLACK o link para a raiz. */
```

Rotações

O código de inserção (= `put()`) é complicado; ele depende de operações de rotação.

Durante uma operação de inserção, podemos ter, temporariamente, um link rubro inclinado para a direita ou dois links rubros incidindo no mesmo nó.

Para corrigir isso, usamos rotações e *flipping colors*.

Rotação esquerda (ou anti-horária) em torno de um nó `h`: o filho direito de `h` "sobe" e adota `h` como seu filho esquerdo.

Continuamos tendo uma `BST` com os mesmos nós, mas raiz diferente.