

Red-Black Trees

Red-Black Trees

balancing binary search trees
relation with 2-3-4 trees

Insertion into a Red-Black Tree

algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

1 Red-Black Trees

balancing binary search trees
relation with 2-3-4 trees

2 Insertion into a Red-Black Tree

algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

MCS 360 Lecture 34
Introduction to Data Structures
Jan Vershelde, 10 November 2010

Red-Black Trees

Red-Black Trees

balancing binary
search trees

relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

an elaborate
example of an insert

pseudo code for a
recursive insert
function

- 1 Red-Black Trees
balancing binary search trees
relation with 2-3-4 trees

- 2 Insertion into a Red-Black Tree
algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

Binary Search Trees

Red-Black Trees

balancing binary
search trees

relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

an elaborate
example of an insert

pseudo code for a
recursive insert
function

Binary search trees store ordered data.

Rules to insert x at node N :

- if N is empty, then put x in N
- if $x < N$, insert x to the left of N
- if $x > N$, insert x to the right of N

Balanced tree of n elements has depth is $\log_2(n)$

\Rightarrow retrieval is $O(\log_2(n))$ or quasi linear.

With rotation we make a tree balanced.

Alternative to AVL tree: nodes with > 2 children.

Every node in a binary tree has at most 2 children.

A 2-3-4 tree has nodes with 2, 3, and 4 children.

A red-black tree is a binary tree equivalent to a 2-3-4 tree.

Binary Search Trees

Red-Black Trees

balancing binary
search trees

relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

an elaborate
example of an insert

pseudo code for a
recursive insert
function

Binary search trees store ordered data.

Rules to insert x at node N :

- if N is empty, then put x in N
- if $x < N$, insert x to the left of N
- if $x > N$, insert x to the right of N

Balanced tree of n elements has depth is $\log_2(n)$

\Rightarrow retrieval is $O(\log_2(n))$ or quasi linear.

With rotation we make a tree balanced.

Alternative to AVL tree: nodes with > 2 children.

Every node in a binary tree has at most 2 children.

A 2-3-4 tree has nodes with 2, 3, and 4 children.

A red-black tree is a binary tree equivalent to a 2-3-4 tree.

Binary Search Trees

Red-Black Trees

balancing binary
search trees

relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

an elaborate
example of an insert

pseudo code for a
recursive insert
function

Binary search trees store ordered data.

Rules to insert x at node N :

- if N is empty, then put x in N
- if $x < N$, insert x to the left of N
- if $x > N$, insert x to the right of N

Balanced tree of n elements has depth is $\log_2(n)$

\Rightarrow retrieval is $O(\log_2(n))$ or quasi linear.

With rotation we make a tree balanced.

Alternative to AVL tree: nodes with > 2 children.

Every node in a binary tree has at most 2 children.

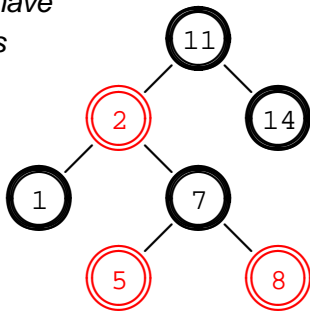
A 2-3-4 tree has nodes with 2, 3, and 4 children.

A red-black tree is a binary tree equivalent to a 2-3-4 tree.

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

a red-black tree

*red nodes have
hollow rings*



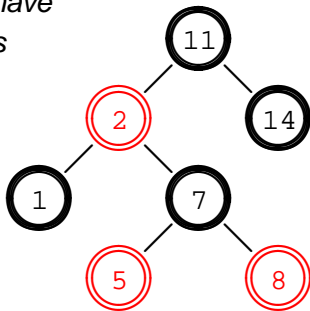
Four invariants:

- 1 A node is either red or black.
- 2 The root is always black.
- 3 A red node has always black children.
- 4 #black nodes in path from root to any leaf is the same.

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

a red-black tree

*red nodes have
hollow rings*



Four invariants:

- 1 A node is either red or black.
- 2 The root is always black.
- 3 A red node has always black children.
- 4 #black nodes in path from root to any leaf is the same.

performance

#black nodes in path from root to any leaf is the same
⇒ equivalent 2-3-4 tree is balanced.

Performance of a red-black tree:

- upper limit of depth of red-black tree: $2 \log_2(n) + 2$
for a search of tree with n elements.
- average cost of search is empirically: $1.002 \log_2(n)$.

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

#black nodes in path from root to any leaf is the same
⇒ equivalent 2-3-4 tree is balanced.

Performance of a red-black tree:

- upper limit of depth of red-black tree: $2 \log_2(n) + 2$
for a search of tree with n elements.
- average cost of search is empirically: $1.002 \log_2(n)$.

Red-Black Trees

Red-Black Trees

balancing binary
search trees

relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

an elaborate
example of an insert

pseudo code for a
recursive insert
function

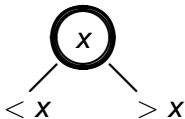
1 Red-Black Trees

balancing binary search trees
relation with 2-3-4 trees

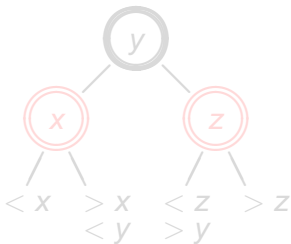
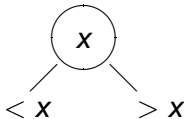
2 Insertion into a Red-Black Tree

algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

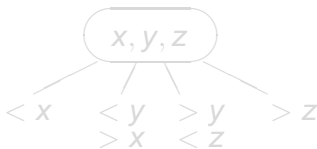
relation with 2-3-4 trees

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

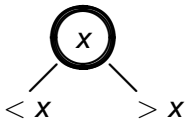
a 2-node



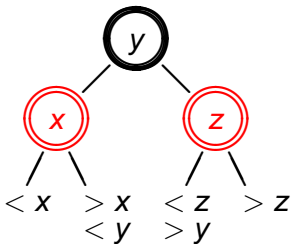
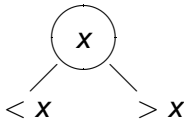
a 4-node



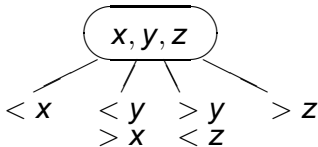
relation with 2-3-4 trees

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

a 2-node

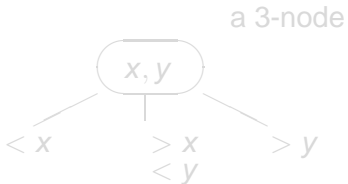
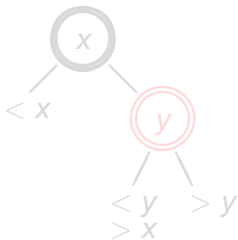
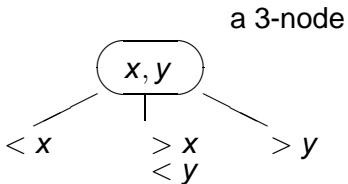
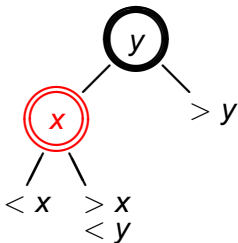


a 4-node



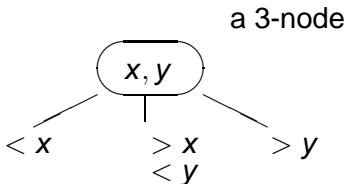
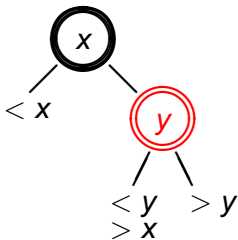
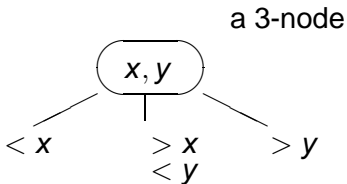
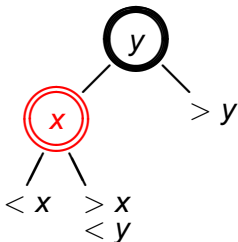
10 Nov 2010

relation with 2-3-4 trees

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

10 Nov 2010

relation with 2-3-4 trees

Red-Black
Treesbalancing binary
search treesrelation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

Red-Black Trees

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

**algorithm for
insertion**
an elaborate
example of an insert
pseudo code for a
recursive insert
function

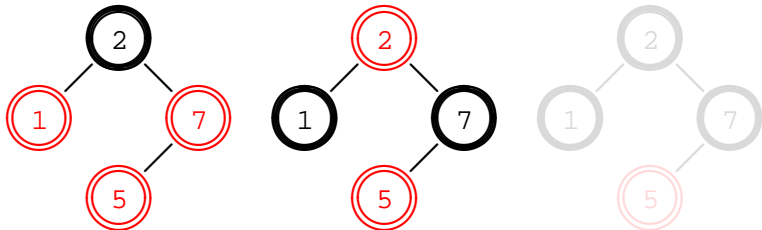
1 Red-Black Trees
balancing binary search trees
relation with 2-3-4 trees

2 Insertion into a Red-Black Tree
algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

Changing Colors

Case 0: The color of a new leaf is always red.
If parent of new leaf is black, then done.

Case 1: We just change colors, e.g.:



- 1 If sibling of parent also red, change color of parent and its sibling to black, and change color of the grandparent.
- 2 Ensure color of the root is black.

Changing Colors

Red-Black Trees

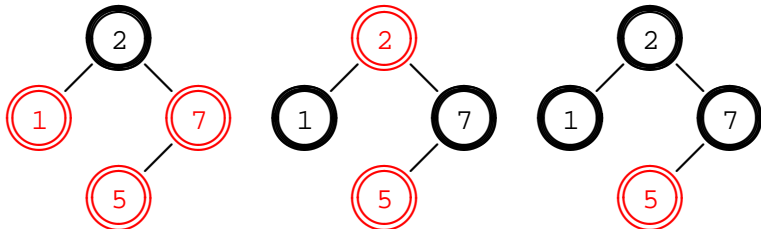
balancing binary search trees
relation with 2-3-4 trees

Insertion into a Red-Black Tree

algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

Case 0: The color of a new leaf is always red.
If parent of new leaf is black, then done.

Case 1: We just change colors, e.g.:



- 1 If sibling of parent also red, change color of parent and its sibling to black, and change color of the grandparent.
- 2 Ensure color of the root is black.

Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

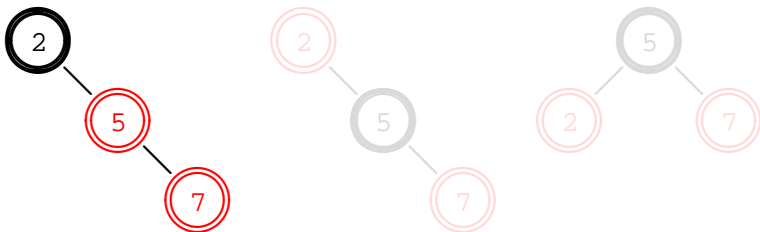
Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

Tree Rotation

What if parent does not have a red sibling?

Case 2: One rotation suffices, e.g.:



- 1 Change the color of the parent to black, change the color of the grandparent to red.
- 2 Rotate to restore the 4th invariant.

Tree Rotation

Red-Black Trees

balancing binary search trees

relation with 2-3-4 trees

Insertion into a Red-Black Tree

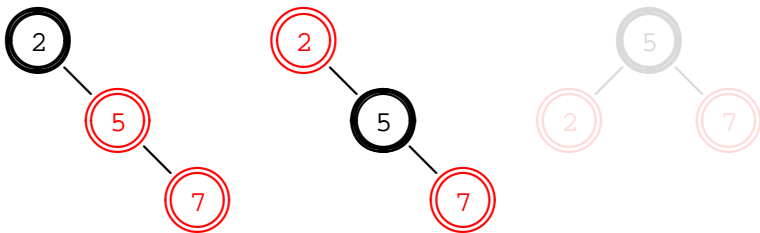
algorithm for insertion

an elaborate example of an insert

pseudo code for a recursive insert function

What if parent does not have a red sibling?

Case 2: One rotation suffices, e.g.:



- 1 Change the color of the parent to black, change the color of the grandparent to red.
- 2 Rotate to restore the 4th invariant.

Tree Rotation

Red-Black Trees

balancing binary search trees

relation with 2-3-4 trees

Insertion into a Red-Black Tree

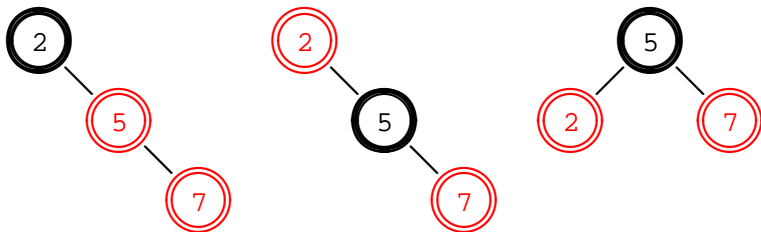
algorithm for insertion

an elaborate example of an insert

pseudo code for a recursive insert function

What if parent does not have a red sibling?

Case 2: One rotation suffices, e.g.:

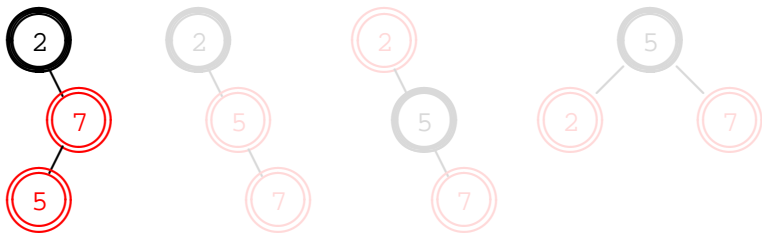


- 1 Change the color of the parent to black, change the color of the grandparent to red.
- 2 Rotate to restore the 4th invariant.

Double Rotation

One rotation works for right-right or left-left tree.

Case 3: Tree is right-left (or left-right), e.g.:



- 1 Rotate right-left tree to right-right tree.
- 2 Change color of parent and grandparent.
- 3 Rotate to restore the 4th invariant.

Double Rotation

Red-Black Trees

balancing binary search trees

relation with 2-3-4 trees

Insertion into a Red-Black Tree

algorithm for insertion

an elaborate example of an insert

pseudo code for a recursive insert function

One rotation works for right-right or left-left tree.

Case 3: Tree is right-left (or left-right), e.g.:

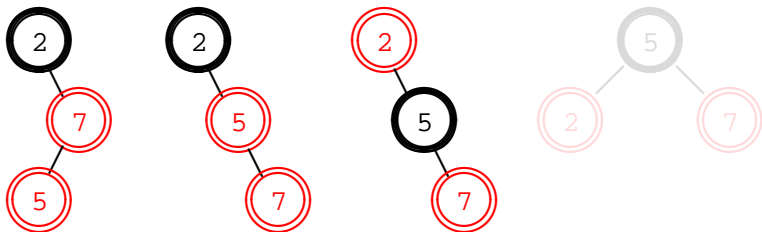


- 1 Rotate right-left tree to right-right tree.
- 2 Change color of parent and grandparent.
- 3 Rotate to restore the 4th invariant.

Double Rotation

One rotation works for right-right or left-left tree.

Case 3: Tree is right-left (or left-right), e.g.:



- 1 Rotate right-left tree to right-right tree.
- 2 Change color of parent and grandparent.
- 3 Rotate to restore the 4th invariant.

Double Rotation

Red-Black Trees

balancing binary search trees

relation with 2-3-4 trees

Insertion into a Red-Black Tree

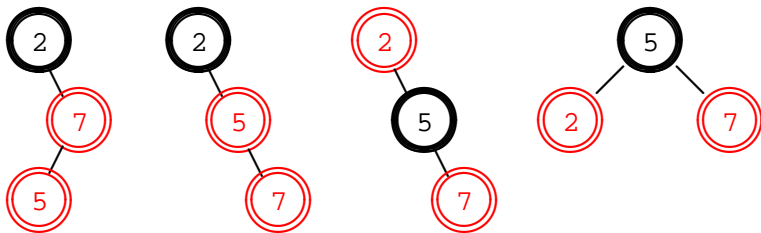
algorithm for insertion

an elaborate example of an insert

pseudo code for a recursive insert function

One rotation works for right-right or left-left tree.

Case 3: Tree is right-left (or left-right), e.g.:



- 1 Rotate right-left tree to right-right tree.
- 2 Change color of parent and grandparent.
- 3 Rotate to restore the 4th invariant.

Red-Black Trees

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
**an elaborate
example of an insert**
pseudo code for a
recursive insert
function

- 1 Red-Black Trees
balancing binary search trees
relation with 2-3-4 trees

- 2 Insertion into a Red-Black Tree
algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

inserting into a red-black tree

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion

**an elaborate
example of an insert**

pseudo code for a
recursive insert
function

During insertion, many cases may occur:

- 1 Locate the new leaf and color it red.
- 2 If in case 0, then done.
- 3 Apply cases 1, 2, and 3, restoring invariants.
 - 1 case 1: parent has red sibling
⇒ change colors
 - 2 case 2: left-left or right-right tree
⇒ change colors and rotate
 - 3 case 3: right-left or left-right tree
⇒ rotate, change colors, rotate

inserting into a red-black tree

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

During insertion, many cases may occur:

- 1 Locate the new leaf and color it red.
- 2 If in case 0, then done.
- 3 Apply cases 1, 2, and 3, restoring invariants.
 - 1 case 1: parent has red sibling
⇒ change colors
 - 2 case 2: left-left or right-right tree
⇒ change colors and rotate
 - 3 case 3: right-left or left-right tree
⇒ rotate, change colors, rotate

inserting into a red-black tree

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

During insertion, many cases may occur:

- 1 Locate the new leaf and color it red.
- 2 If in case 0, then done.
- 3 Apply cases 1, 2, and 3, restoring invariants.
 - 1 case 1: parent has red sibling
⇒ change colors
 - 2 case 2: left-left or right-right tree
⇒ change colors and rotate
 - 3 case 3: right-left or left-right tree
⇒ rotate, change colors, rotate

inserting into a red-black tree

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

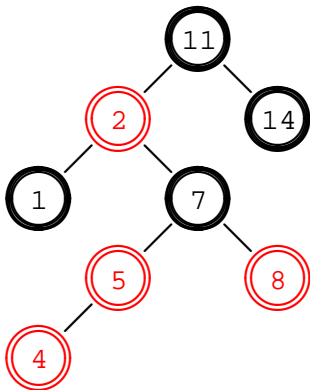
Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

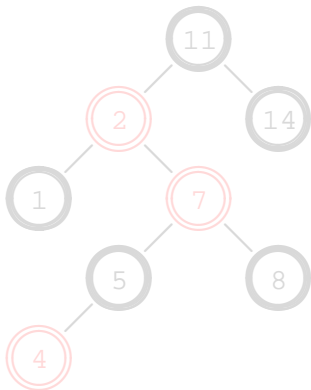
During insertion, many cases may occur:

- 1 Locate the new leaf and color it red.
- 2 If in case 0, then done.
- 3 Apply cases 1, 2, and 3, restoring invariants.
 - 1 case 1: parent has red sibling
⇒ change colors
 - 2 case 2: left-left or right-right tree
⇒ change colors and rotate
 - 3 case 3: right-left or left-right tree
⇒ rotate, change colors, rotate

10 Nov 2010

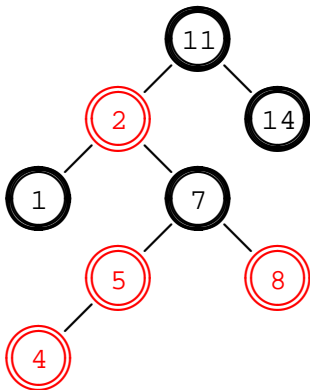
Red-Black
Treesbalancing binary
search trees
relation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertion
**an elaborate
example of an insert**
pseudo code for a
recursive insert
function

inserting 4

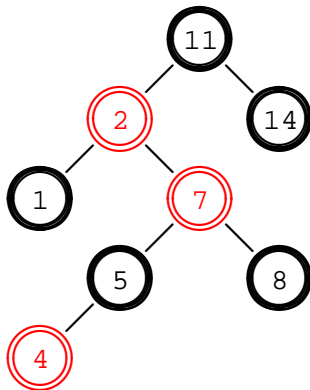


After changing colors of 5 and 8 to black,
and changing 7 to red, but then parent of 7 is red too...

10 Nov 2010

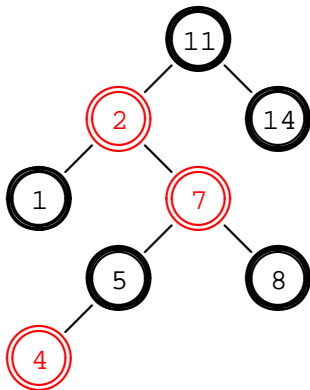
Red-Black
Treesbalancing binary
search trees
relation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertionan elaborate
example of an insertpseudo code for a
recursive insert
function

inserting 4

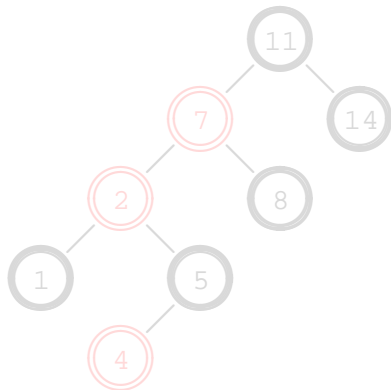


After changing colors of 5 and 8 to black,
and changing 7 to red, but then parent of 7 is red too...

10 Nov 2010

Red-Black
Treesbalancing binary
search trees
relation with 2-3-4
treesInsertion into
a Red-Black
Treealgorithm for
insertion
**an elaborate
example of an insert**
pseudo code for a
recursive insert
function

rotating around 2



Case 3: rotate left-right tree (11-2-7) to left-left tree
is the first step...

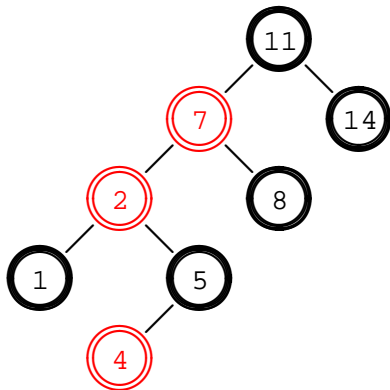
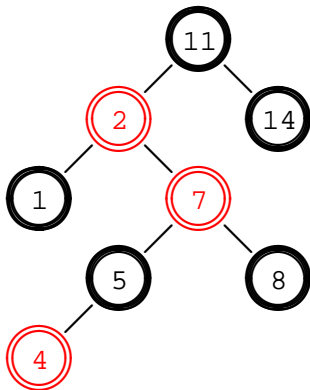
Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

rotating around 2



Case 3: rotate left-right tree (11-2-7) to left-left tree
is the first step...

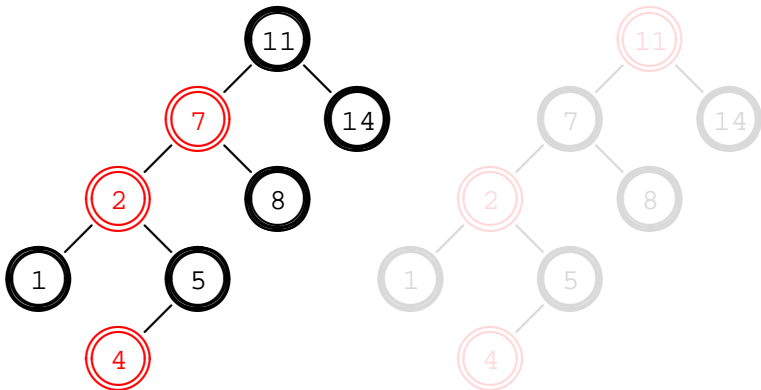
Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
**an elaborate
example of an insert**
pseudo code for a
recursive insert
function

changing colors



The second step in Case 3 is to change the color of 7 to black and the color of 11 to red.

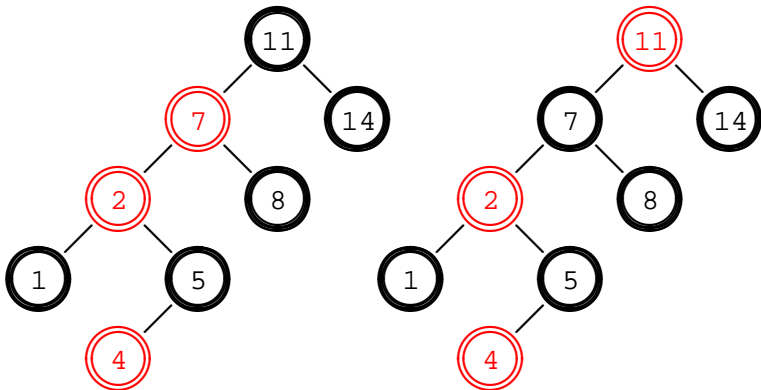
Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

changing colors



The second step in Case 3 is to change the color of 7 to black and the color of 11 to red.

Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

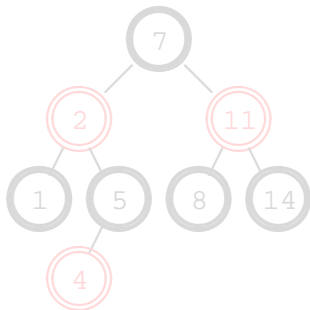
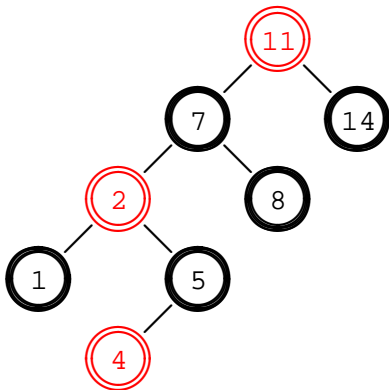
Insertion into
a Red-Black
Tree

algorithm for
insertion

**an elaborate
example of an insert**

pseudo code for a
recursive insert
function

rotate around 11



In the last step of Case 3, we restore the balance by a right rotation around 11.

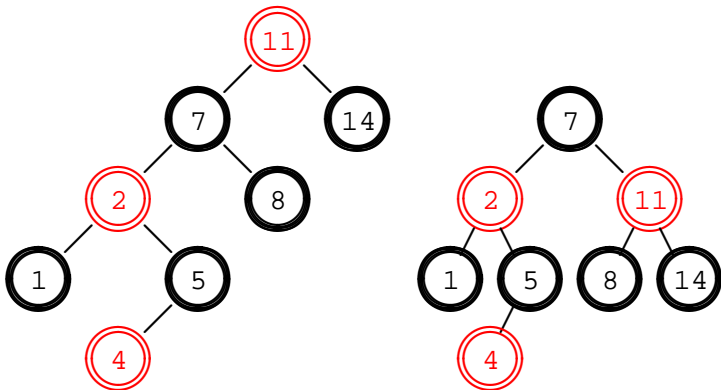
Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

rotate around 11



In the last step of Case 3, we restore the balance by a right rotation around 11.

Red-Black Trees

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert

**pseudo code for a
recursive insert
function**

- 1 Red-Black Trees
balancing binary search trees
relation with 2-3-4 trees

- 2 Insertion into a Red-Black Tree
algorithm for insertion
an elaborate example of an insert
pseudo code for a recursive insert function

Recursive Insert Function

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert

pseudo code for a
recursive insert
function

We extend the node type of a binary search tree:

- 1 a node contains data (often an integer key);
- 2 in addition, every node has a color: red or black;
- 3 two pointers to nodes lead to left and right children.

Prototype of a recursive insert function:

```
bool insert ( Tree &root, Item_Type item );  
  
// returns false if item belonged to root  
  
// returns true if item was added to the root
```

Recursive Insert Function

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert

pseudo code for a
recursive insert
function

We extend the node type of a binary search tree:

- 1 a node contains data (often an integer key);
- 2 in addition, every node has a color: red or black;
- 3 two pointers to nodes lead to left and right children.

Prototype of a recursive insert function:

```
bool insert ( Tree &root, Item_Type item );  
  
// returns false if item belonged to root  
  
// returns true if item was added to the root
```

Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert

**pseudo code for a
recursive insert
function**

```
bool insert ( Tree &root, Item_Type item )
    if(root == NULL)
        root = new black node;
        return true;
    else if(item == root->data)
        return false;
    else if (item < root->data)
        if(left == NULL)
            left = new red node;
            return true;
        else if((left == red) && (right == red))
            change left and right to black
            and color the local root red;
```

pseudo code continued

Red-Black
Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into
a Red-Black
Tree

algorithm for
insertion
an elaborate
example of an insert
pseudo code for a
recursive insert
function

```

if(insert(left,item))
    if(left grandchild == red)
        change left to black
        and color the local root red;
        rotate the local root right;
    else if(right grandchild == red)
        rotate the left child left;
        change left to black
        and color the local root to red;
        rotate the local root right;

else // item > root->data: exercise

if(local root is root of the tree)
    color the root black.

```

Summary + Assignments

Red-Black Trees

balancing binary
search trees
relation with 2-3-4
trees

Insertion into a Red-Black Tree

algorithm for
insertion
an elaborate
example of an insert

pseudo code for a
recursive insert
function

More on chapter 11 on balancing binary search trees.

Assignments:

- 1 Take the largest example of a red-black tree on these slides and draw the equivalent 2-3-4 tree.
- 2 Write the pseudo code for the case when the item is inserted to the right child.
- 3 Instead of inserting 4 in the elaborate example, insert 9 and illustrate all stages in the insertion.

Homework due Monday 15 November, at noon:

#2, 3 of L-27; #1, 2 of L-28; and #2 of L-29.

Project 4 is due on Monday!