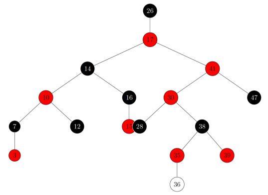


Red-Black Trees 1
Lecture 15

COSC 242 – Algorithms and Data Structures

Today’s outline

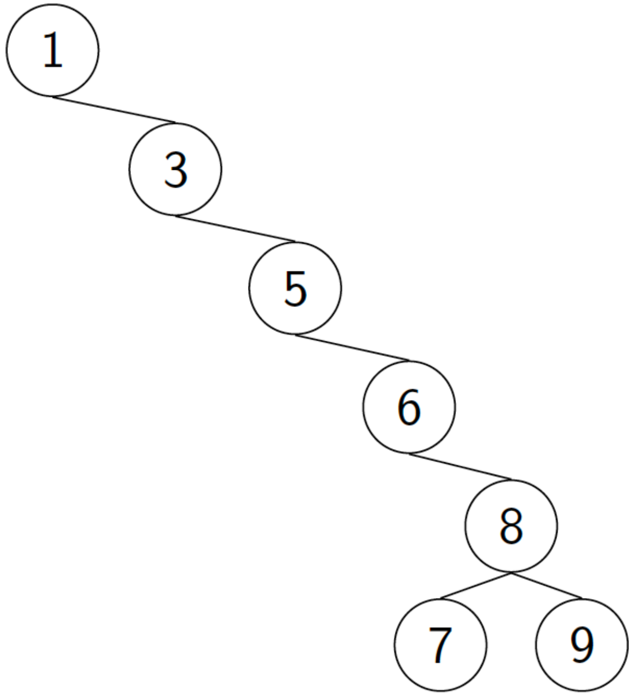
1. Trees and balance
2. Red-black tree properties
3. Black-height definition
4. Number of nodes in an RBT

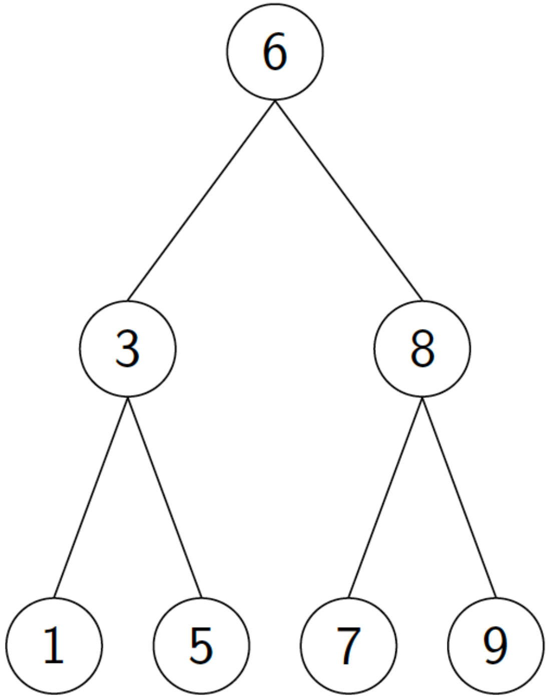
Today’s outline

1. Trees and balance
2. Red-black tree properties
3. Black-height definition
4. Number of nodes in an RBT

The importance of balance

Search, insert, and delete are much more efficient for some BSTs than others. Contrast a BST created by inserting 1,3,5,6,8,7,9 with a second created by inserting 6,3,8,1,5,7,9:





The importance of balance

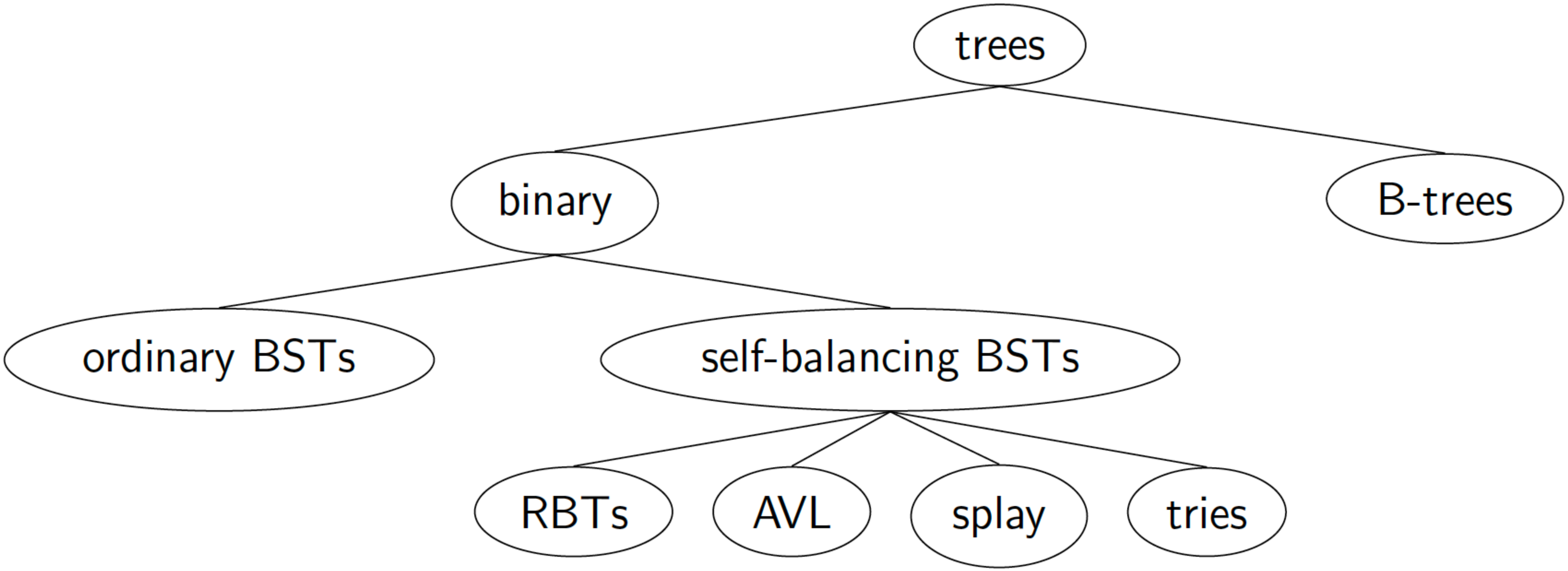
Tall trees are less efficient than short trees. Note that both of these trees have the same keys, so for any given set of keys we would prefer the shortest tree containing those keys.

But building optimal BSTs requires rebuilding the whole tree each time a key is inserted or deleted, and this is very expensive.

Instead, there are several good approximate algorithms that *rebalance* BSTs dynamically.

We will look at one such algorithm: **Red-Black Trees** (RBTs).

A taxonomy of trees



Today's outline

1. Trees and balance
2. Red-black tree properties
3. Black-height definition
4. Number of nodes in an RBT

Overview

Red-black trees are:

- A variation on binary search trees
- Balanced: height is $O(\log n)$, where n is the number of nodes.
- Operations will take $O(\log n)$ time in worst case

Red-black trees

A Red-Black Tree (RBT) is a BST with +1 bit per node: an attribute *colour*, which is either **red** or **black**.

All leaves are empty (nil or null) and coloured black.

- We use a single sentinel, *T.nil*, for all leaves of red-black tree *T*.
- *T.nil.colour* is **black**
- The root's parent is also *T.nil*

All other attributes of BSTs are inherited by red-black trees: key, left, right, parent.

Internal and external nodes

If a child or parent of a node does not exist, the corresponding pointer attribute of the node contains NIL.

We regard these NILs as pointers to leaves, and are **external nodes** of the BST.

Normal key-bearing nodes are **internal nodes** of the tree.

RBT Properties

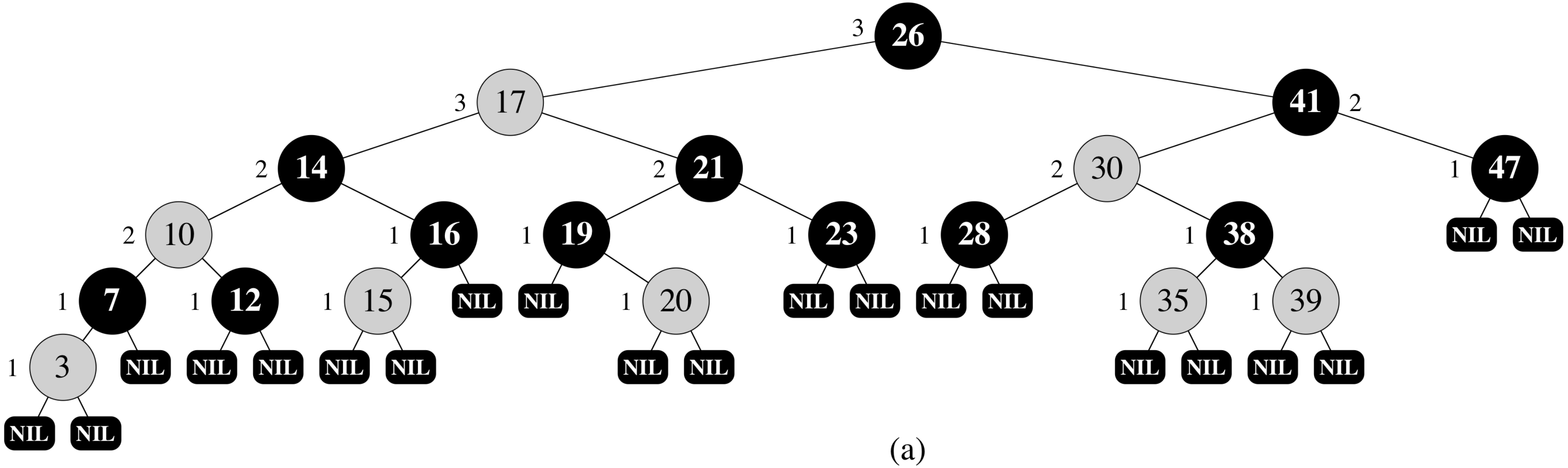
An RBT is a BST with the following properties:

1. Every node is either **red** or **black**.
2. The root is **black**.
3. Every leaf (nil/null) is **black**.
4. If a node is red, then both its children are **black***.
5. For each node, all paths from the node to leaves contain the same number of **black** nodes.

* This implies we cannot have consecutive red nodes.

Visualising an RBT

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to leaves contain the same number of black nodes.

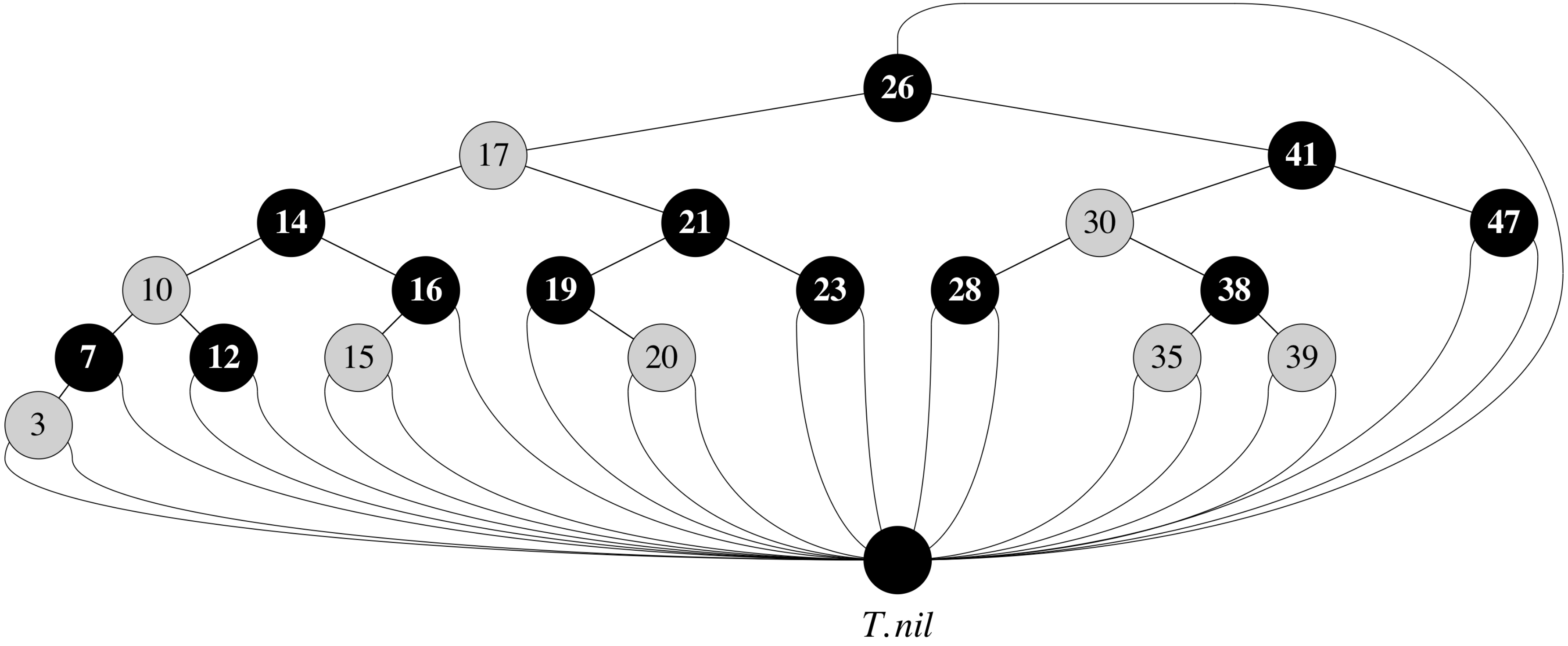


(a)

Legend

- **Black** nodes are darkened, **red** nodes are shaded.
- Every leaf (external nodes), are **black**, and shown as NIL
- Each node is marked with its "black height". E.g., black height of the root is 3.
- Leaves (external nodes) have a black height of 0 (unmarked).

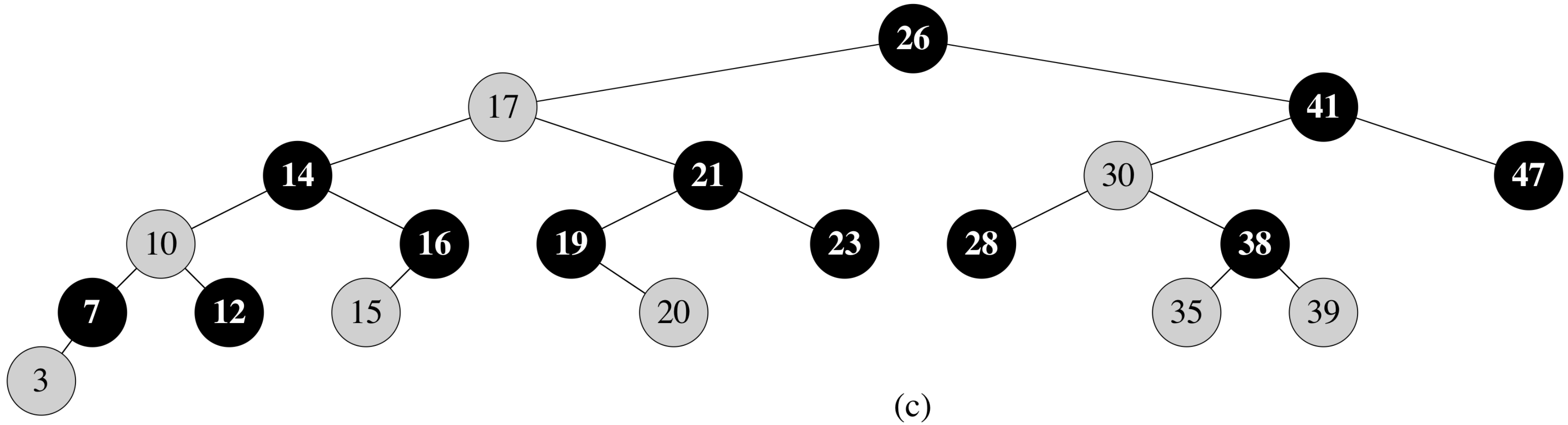
Visualising an RBT



Legend

- The same tree, with each NIL replaced by a single sentinel *T.nil*, which is always black.
- Black heights omitted.

Visualising an RBT

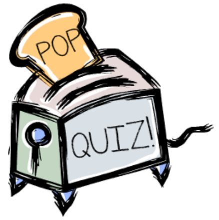


(c)

Legend

- The same tree, with leaves and root's parents (all NILs) omitted.
- This is the visualisation style used throughout the textbook.

Pop quiz 1

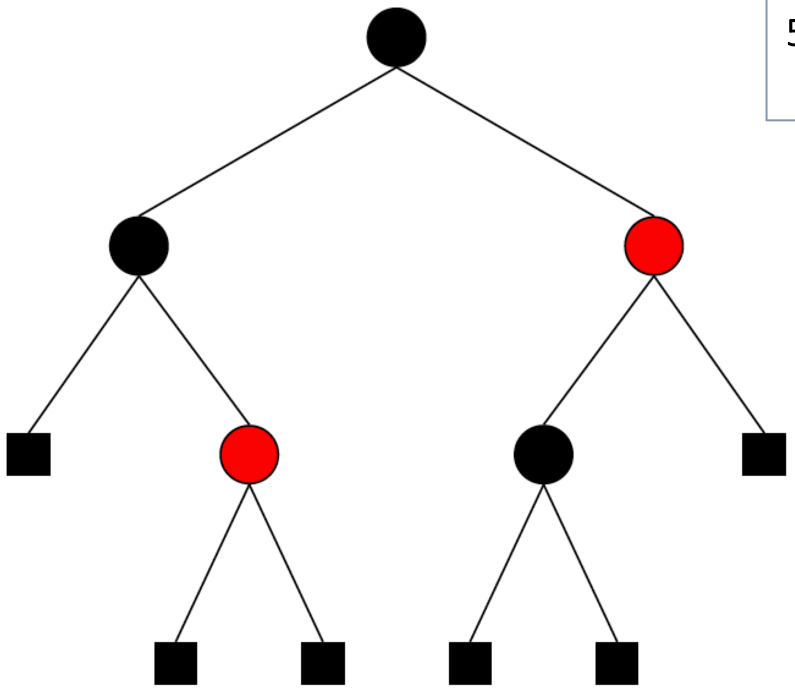


Is this a valid RBT? If not, why not?

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to leaves contain the same number of black nodes.

Answer

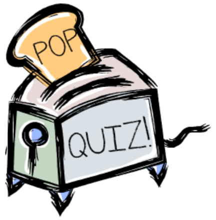
- Yes
- No



Legend

- In in this diagram, rectangles are leaf nodes.

Pop quiz 2

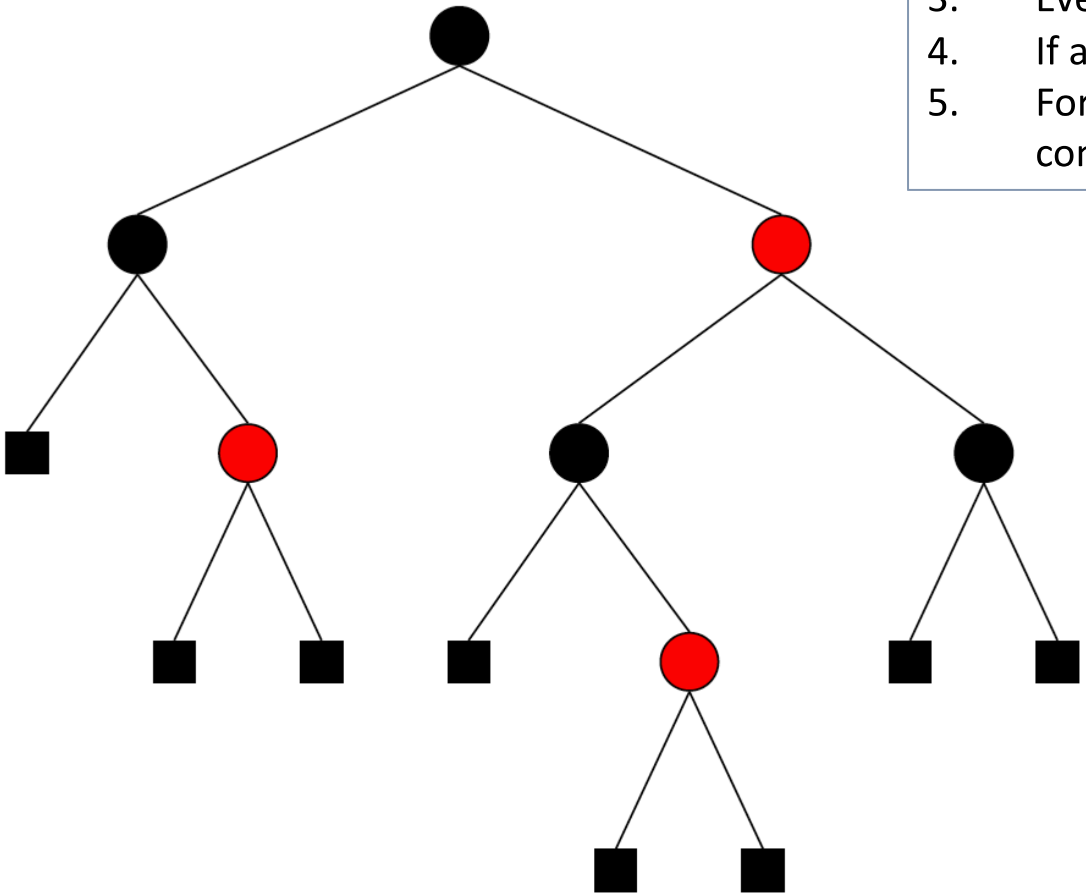


Is this a valid RBT? If not, why not?

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to leaves contain the same number of black nodes.

Answer

- Yes
- No



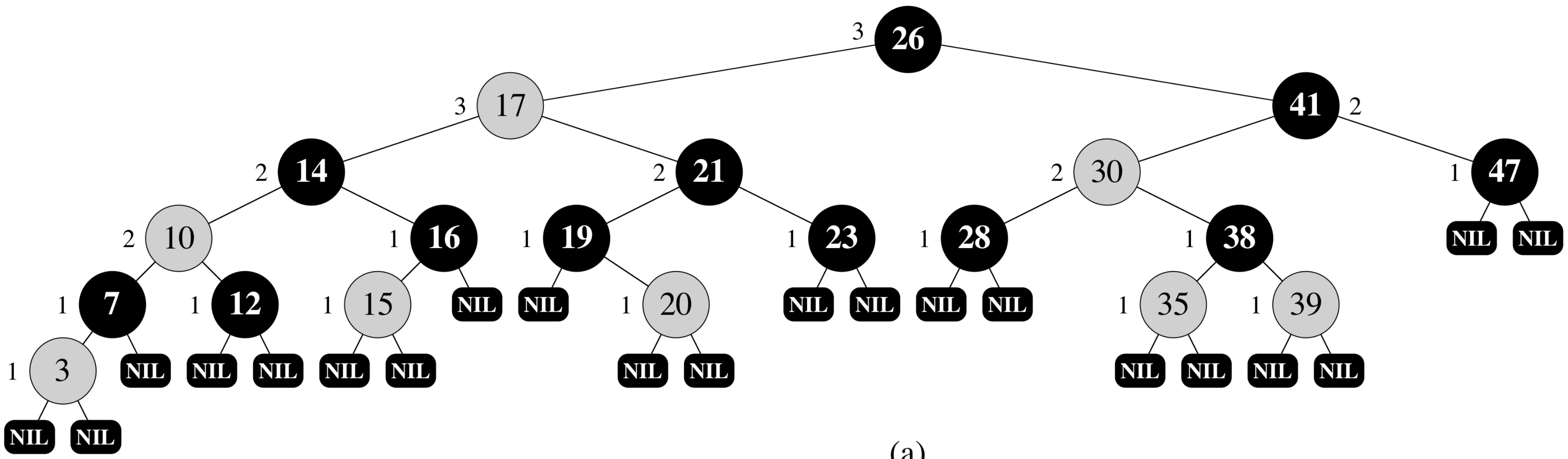
Today’s outline

1. Trees and balance
2. Red-black tree properties
3. Black-height definition
4. Number of nodes in an RBT

Black-height

Definition

The number of **black** nodes on any simple path from, but not including, a node *x* down to a leaf, that also includes the leaf node. Denoted bh(*x*).



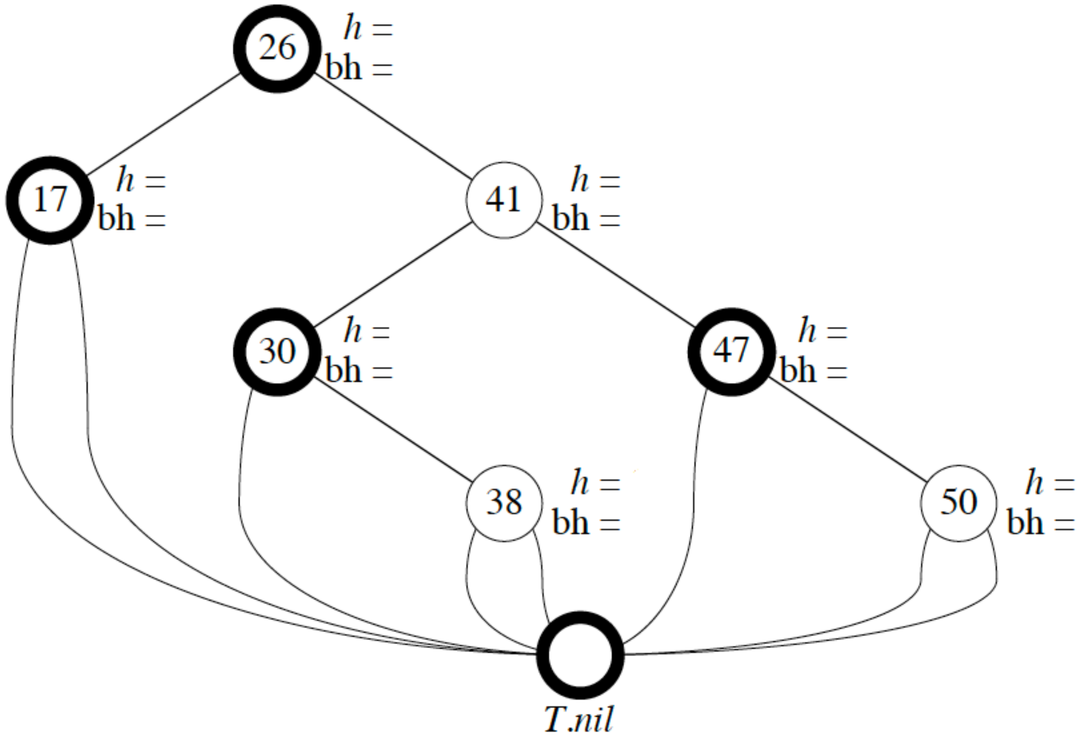
(a)

The black-height of an RBT is the black-height of the root.

Class challenge

Fill in the black-heights (bh) and heights (h) of all the nodes in the RBT.

In this figure, black nodes are indicated by thick bold outlines.



Today’s outline

1. Trees and balance
2. Red-black tree properties
3. Black-height definition
4. Number of nodes in an RBT

Number of internal nodes in an RBT

Lemma

Let n(x) be the number of internal nodes in an RBT rooted at node x, then:

$$n(x) \geq 2^{bh(x)} - 1$$

Proof – by induction

Base case: If bh(x) = 0*, then *x* must be a leaf (T.nil), and n(x) = 0, and:

$$2^{bh(x)} - 1 = 2^0 - 1 = 1 - 1 = 0$$

as required.

* recall: we don't count the current node when calculating bh(x).

Number of nodes in an RBT

Induction: Assume the claim holds for every node of height $h(x) = k$ (induction hypothesis). We need to show the claim is also true for nodes of height $h(x) = k + 1$.

Consider an internal node x that has at least one child of height k. If the child is red, then its black height will be $b_h(y) = b_h(x)$. However, if the colour of the child is black, then its black height must be 1 less than x, so $b_h(y) = b_h(x) - 1$. So, $b_h(y) \geq b_h(x) - 1$.

By the induction hypothesis, each subtree of x has at least:
$2^{b_h(x)-1} - 1$ internal nodes.

Number of nodes in an RBT

So:

+1 for *x* itself — Child 1 — Child 2

$$
\begin{aligned}
n(x) &= 1 + n(y_1) + n(y_2) \\
&\geq 1 + \left(2^{b_h(y_1)} - 1\right) + \left(2^{b_h(y_2)} - 1\right) \\
&\geq 1 + \left(2^{b_h(x)-1} - 1\right) + \left(2^{b_h(x)-1} - 1\right) \\
&\geq 2 \cdot \left(2^{b_h(x)-1} - 1\right) + 1 \\
&\geq 2^{b_h(x)} - 1 \quad \blacksquare
\end{aligned}
$$

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. **If a node is red, then both its children are black.**
5. For each node, all paths from the node to leaves contain the same number of black nodes.

Height of an RBT

Claim

Any node with height h has a black-height ≥ h/2

Proof

By RBT property 4, ≤ h/2 nodes on the path from the node to the leaf are red. Hence, ≥ h/2 are black. ■

Height of an RBT

Theorem

An RBT with n internal nodes has height:*

$$h \le 2\log_2(n+1)$$

* Actual height, not black height

Height of an RBT

Proof

Let h and b_h be the height and black-height of the root, respectively.

From the previous lemma and claim:

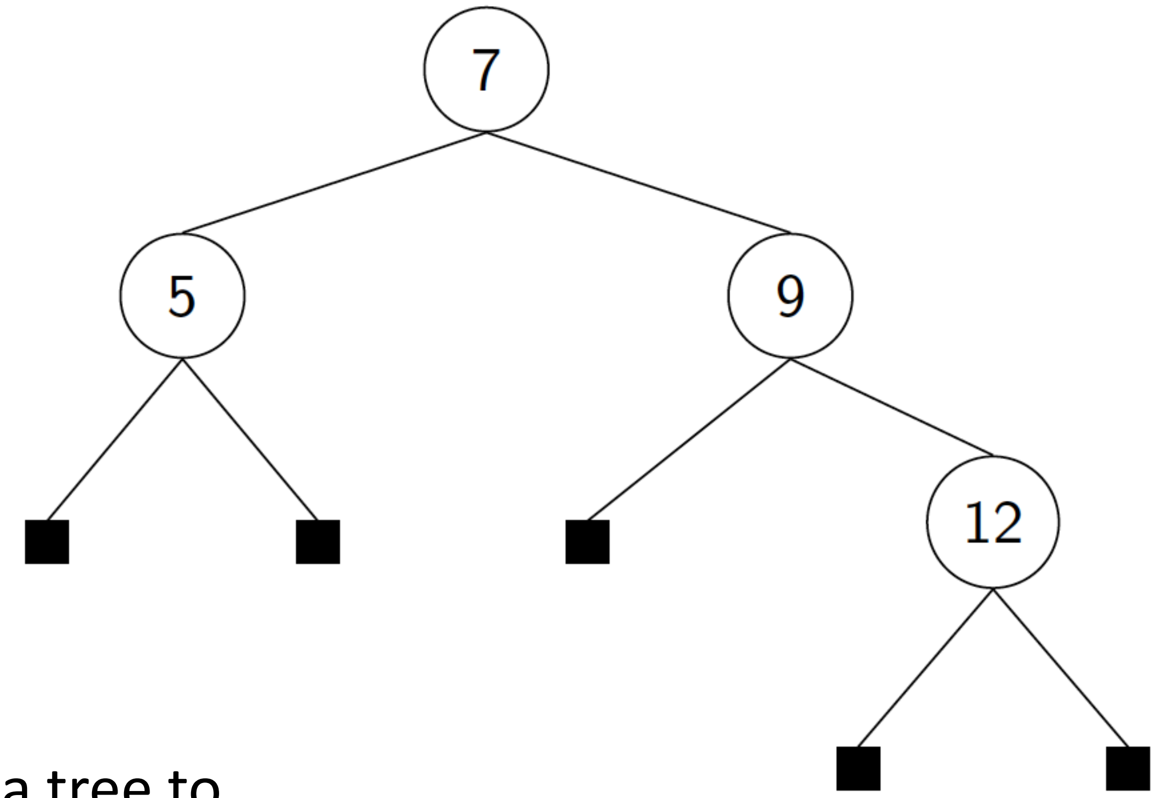
$$
\begin{aligned}
n(x) &\geq 2^{b_h(x)} - 1 \\
&\geq 2^{h(x)/2} - 1
\end{aligned}
$$

Adding one to both sides, then taking logs gives:

$$
\begin{aligned}
\log_2(n(x) + 1) &\geq \log_2\left(2^{h(x)/2}\right) \\
&\geq h(x)/2 \\
h(x) &\geq 2\log_2(n + 1) \qquad \blacksquare
\end{aligned}
$$

Inserting into an RBT

- Insert 8: can we colour the tree to make it an RBT?
- Insert 11
- Insert 10



We need to be able to efficiently rebuild a tree to maintain its RBT qualities.

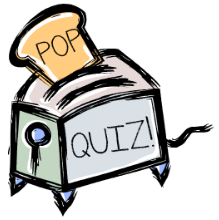
Suggested reading

Red-black trees are the topic of Chapter 13 of the textbook.

Today's lecture covered section 13.1.

Solutions

Pop quiz 1

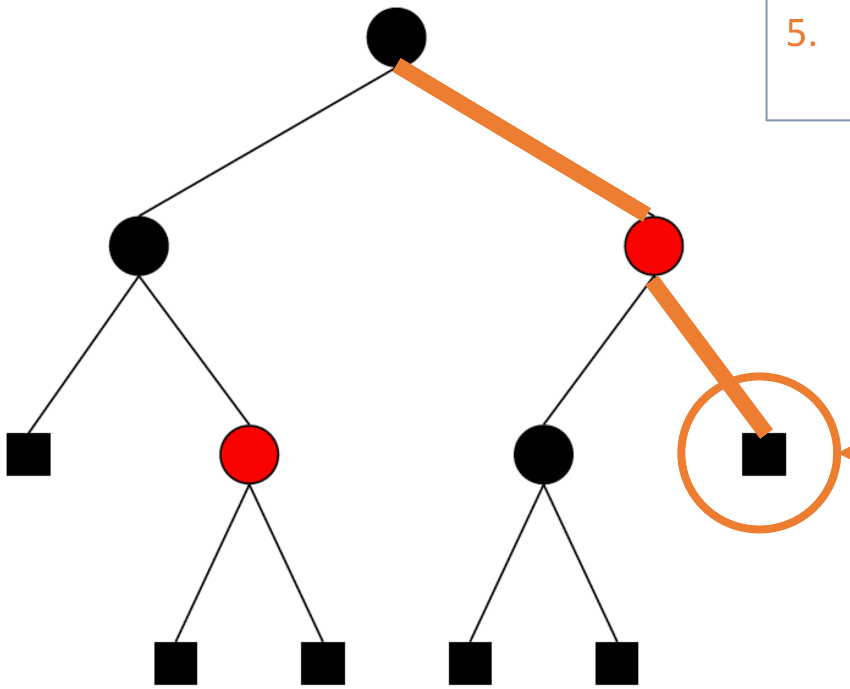


Is this a valid RBT? If not, why not?

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to leaves contain the same number of black nodes.

Answer

- Yes
- No ✓

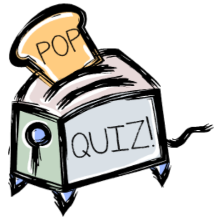


Path to leaf node does not pass through same number of black nodes as other leaves.

Legend

- In in this diagram, rectangles are leaf nodes.

Pop quiz 2

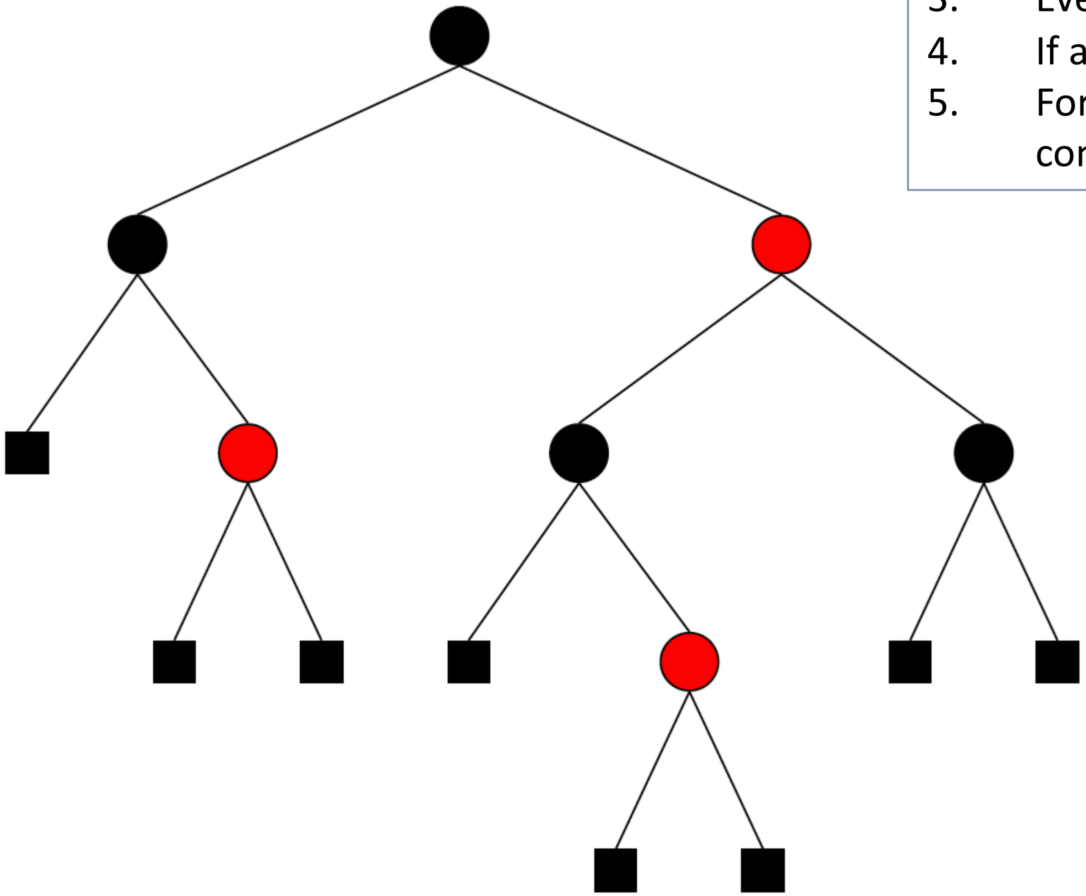


Is this a valid RBT? If not, why not?

1. Every node is either red or black.
2. The root is black.
3. Every leaf (nil/null) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to leaves contain the same number of black nodes.

Answer

- Yes ✓
- No



Class challenge

Activity Time!

Fill in the black-heights (bh) and heights (h) of all the nodes in the RBT:

In this figure, black nodes are indicated by thick bold outlines.

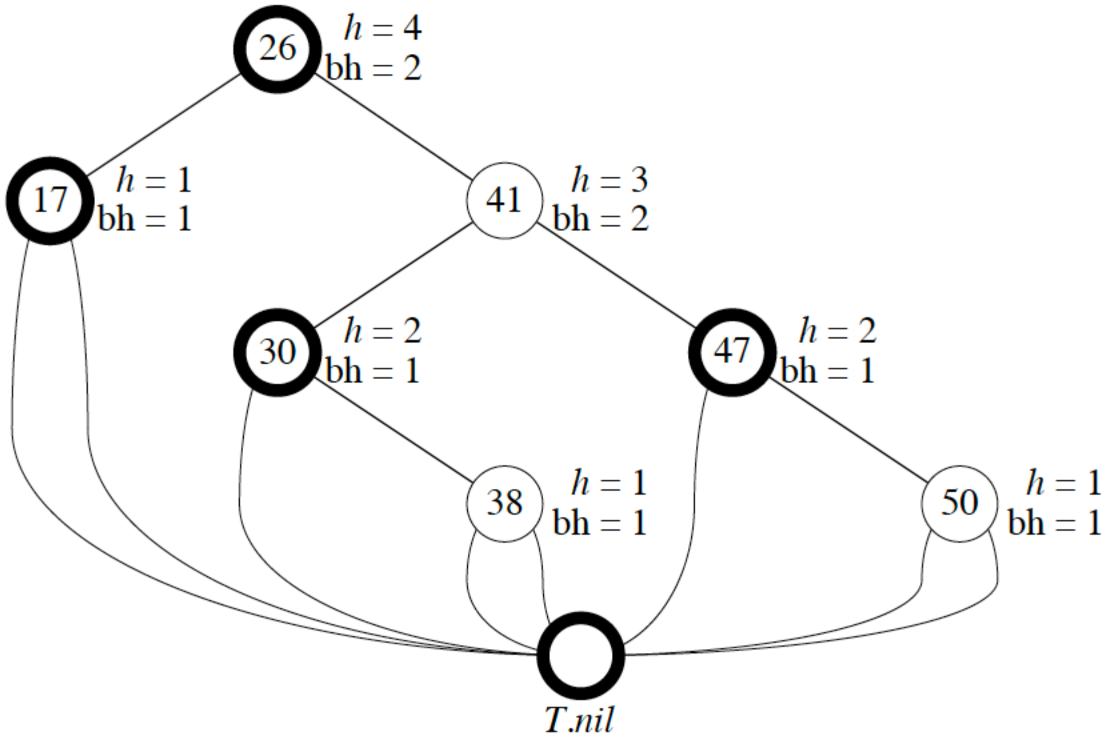


Image attributions

This Photo by Unknown Author is licensed under CC BY

Disclaimer: Images and attribution text provided by PowerPoint search. The author has no connection with, nor endorses, the attributed parties and/or websites listed above.