Lecture 5

Red-Black Trees, Height Bound, Insertion

Source: Introduction to Algorithms, CLRS

RB-Trees: Black Height

RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**,

RB-Trees: Black Height

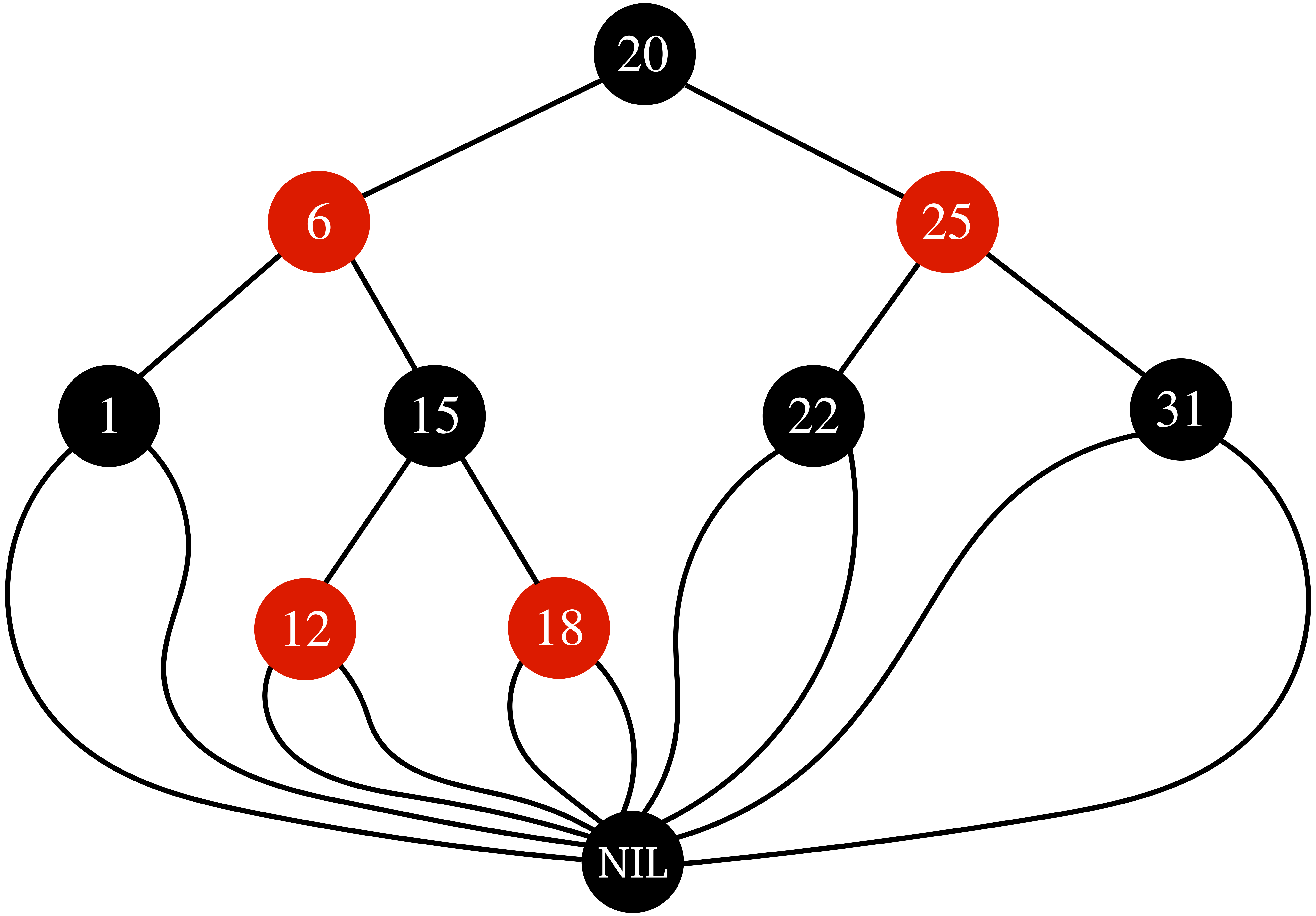
Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x,

RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.

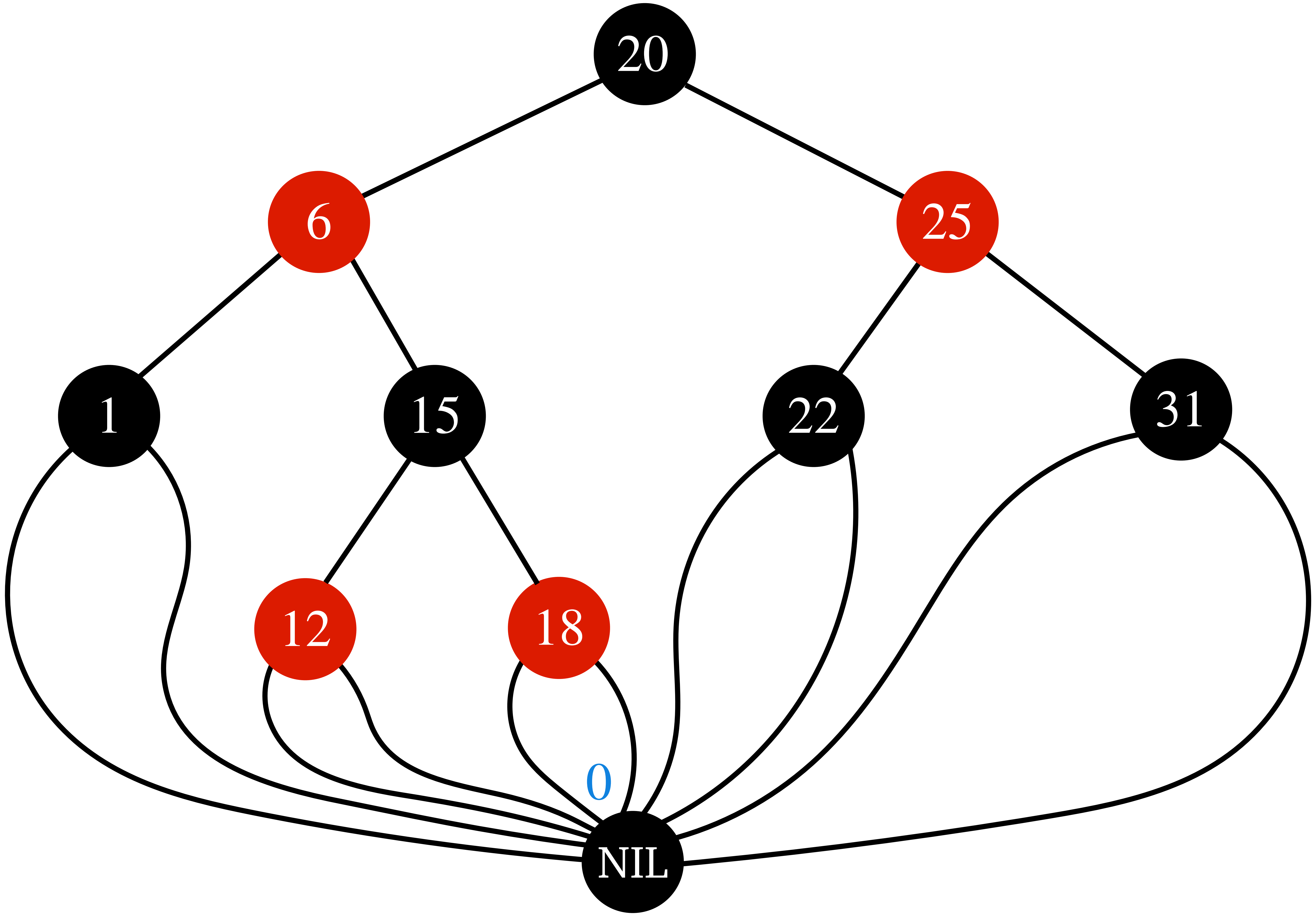
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



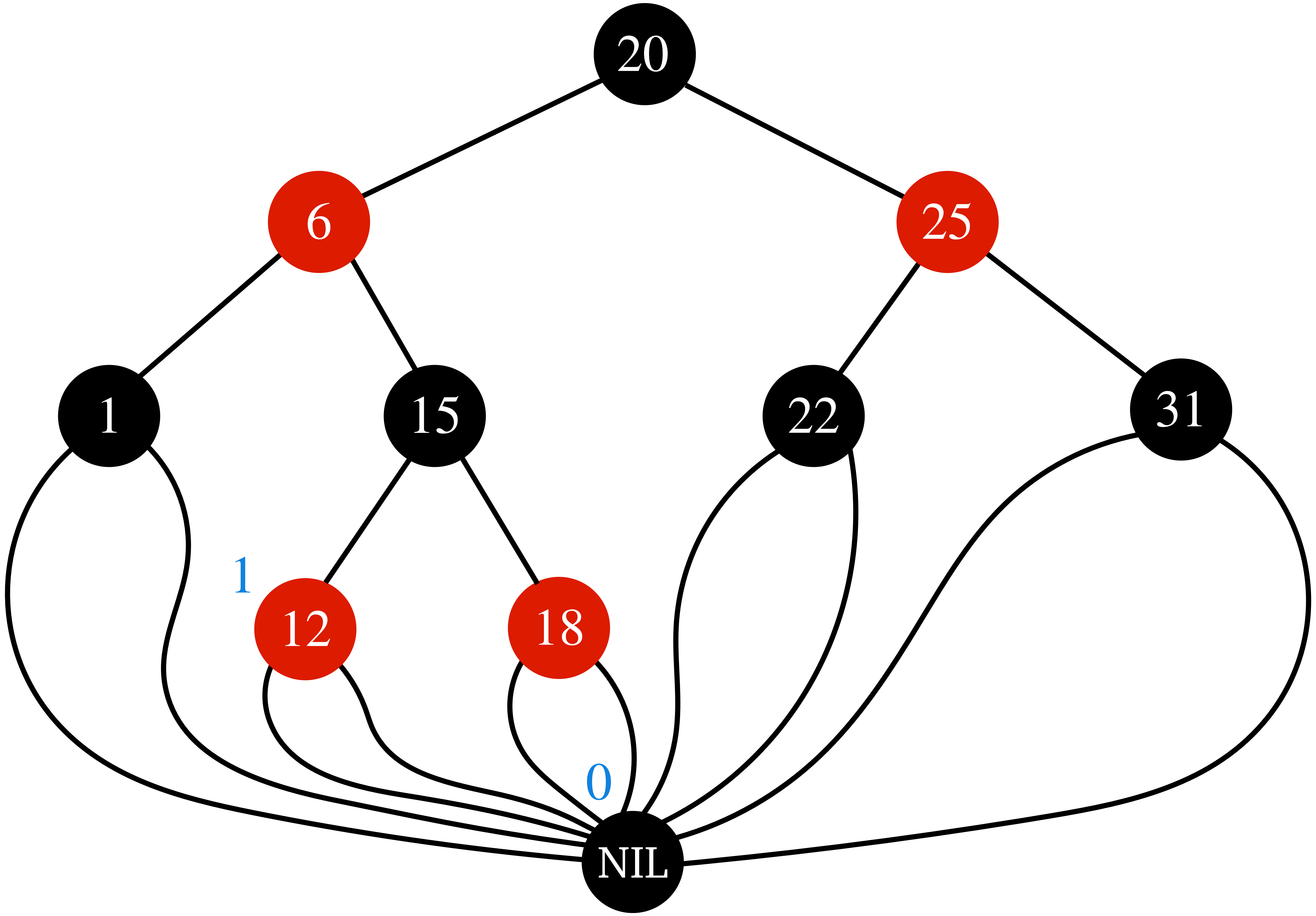
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



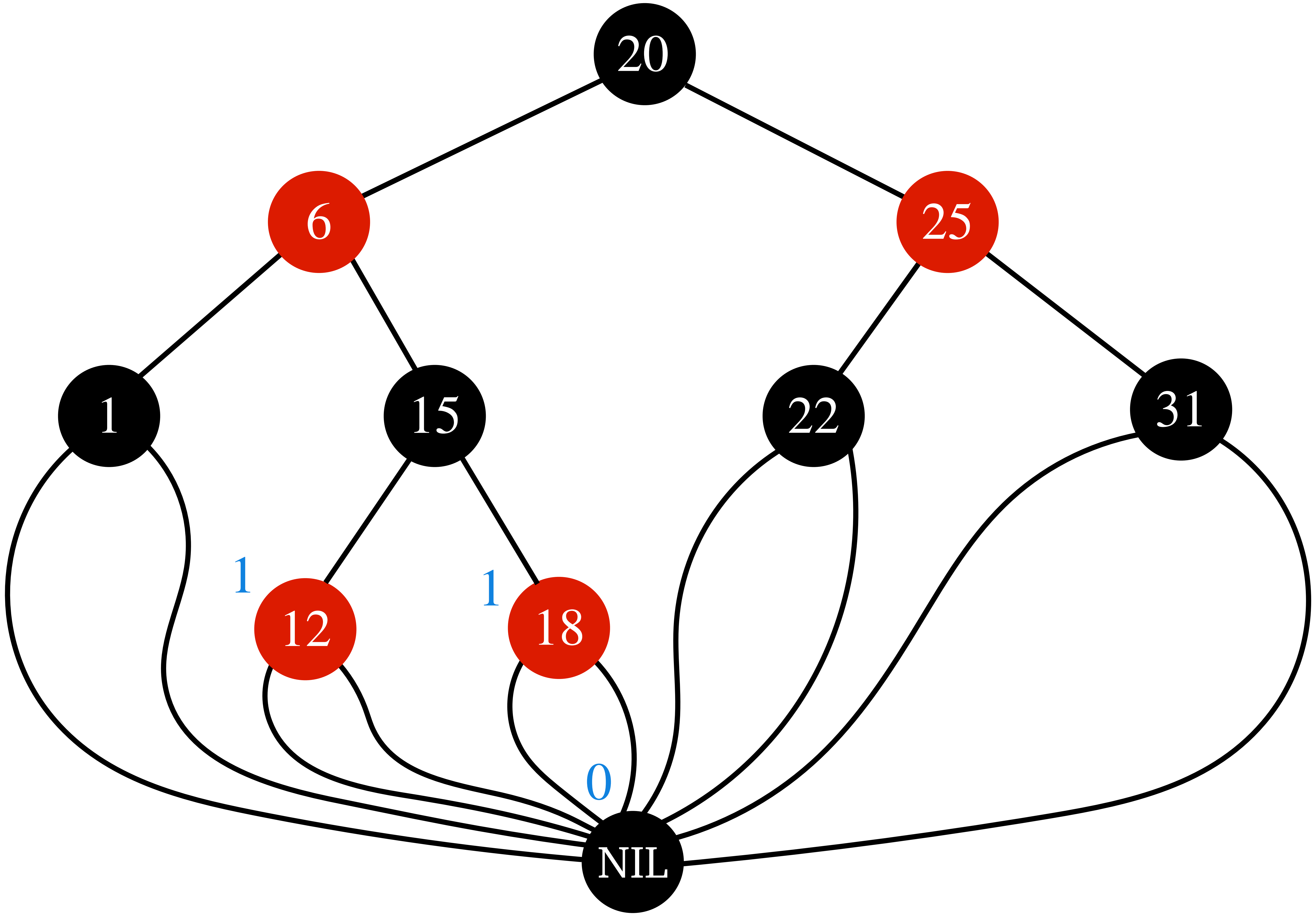
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



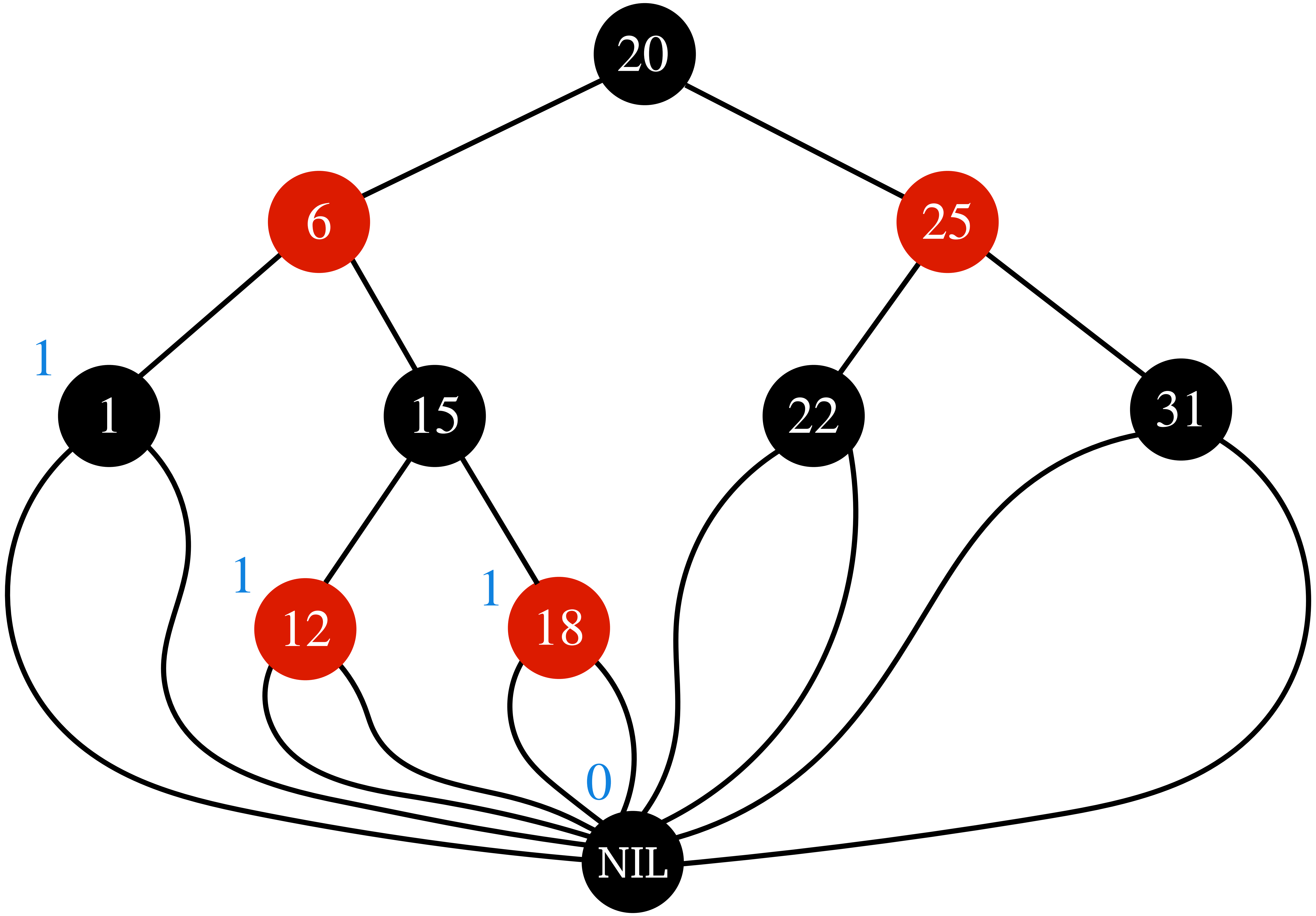
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



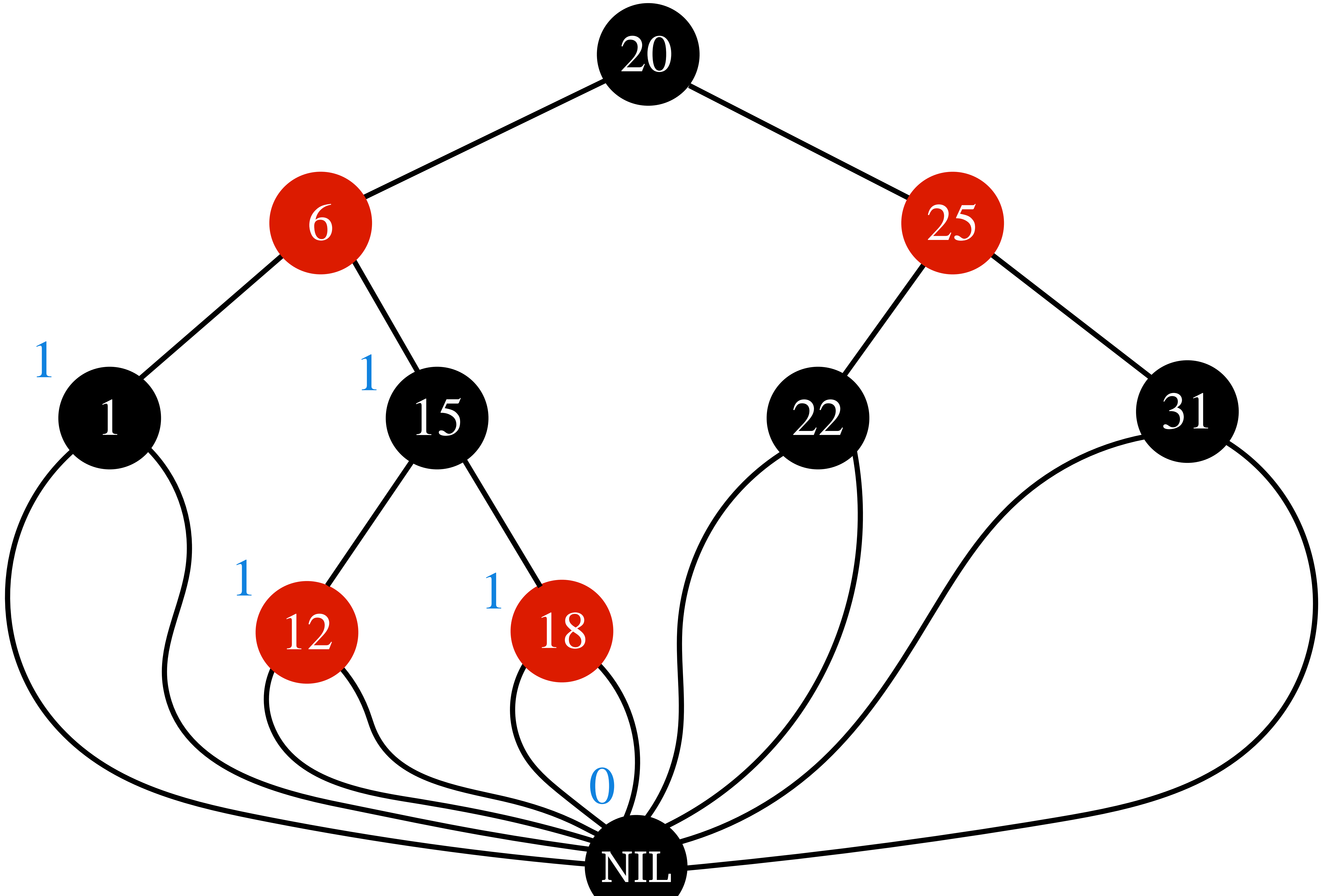
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



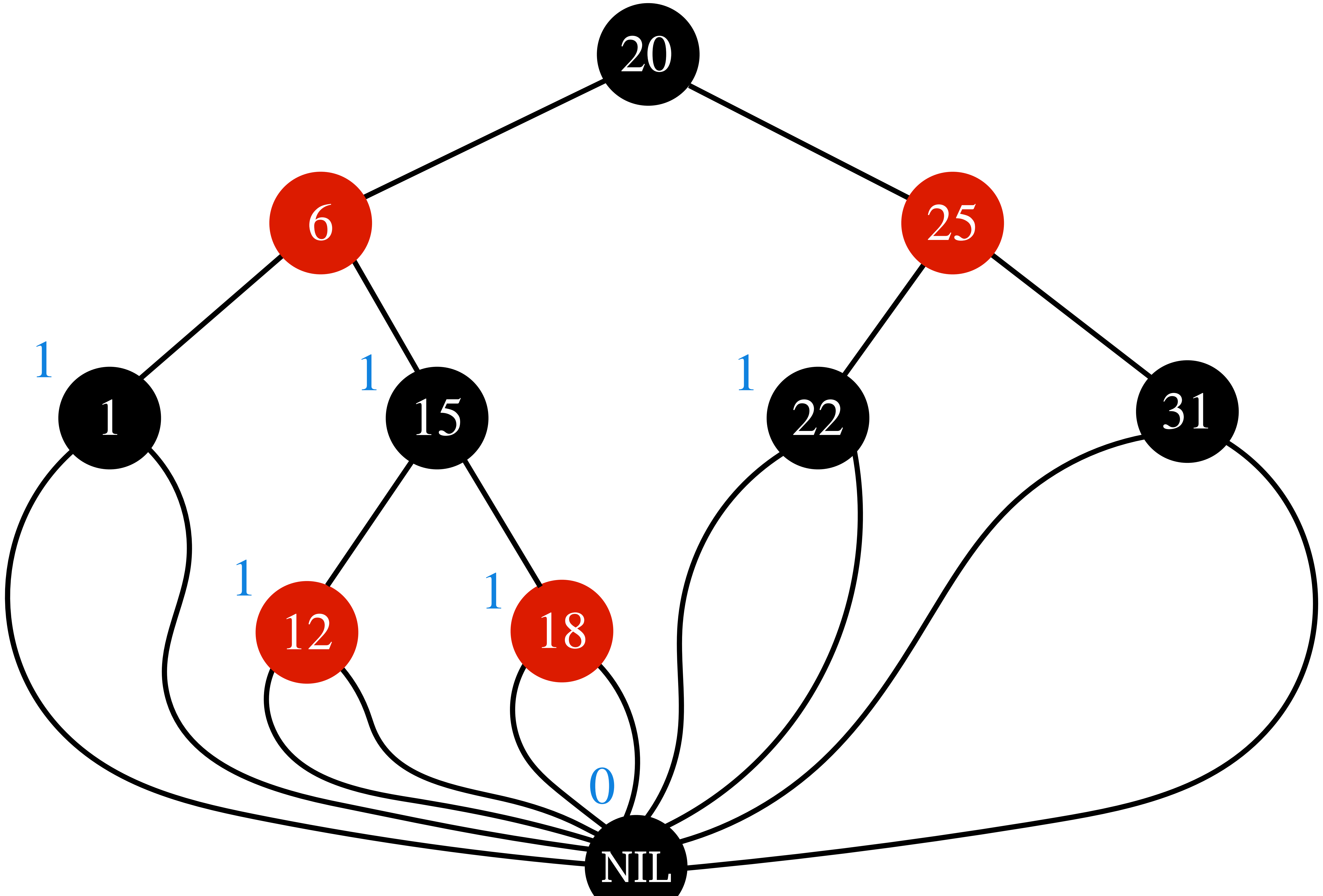
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



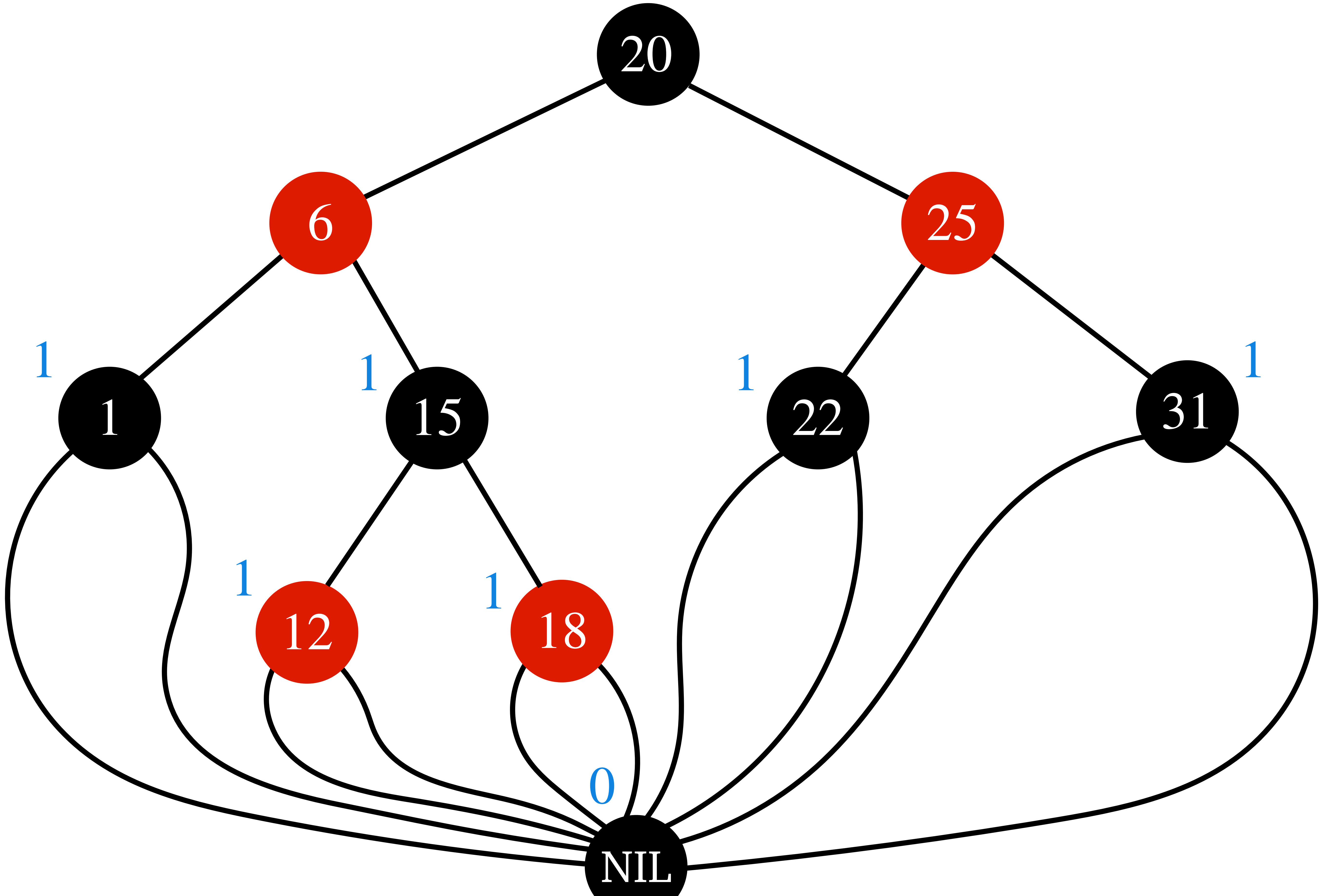
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



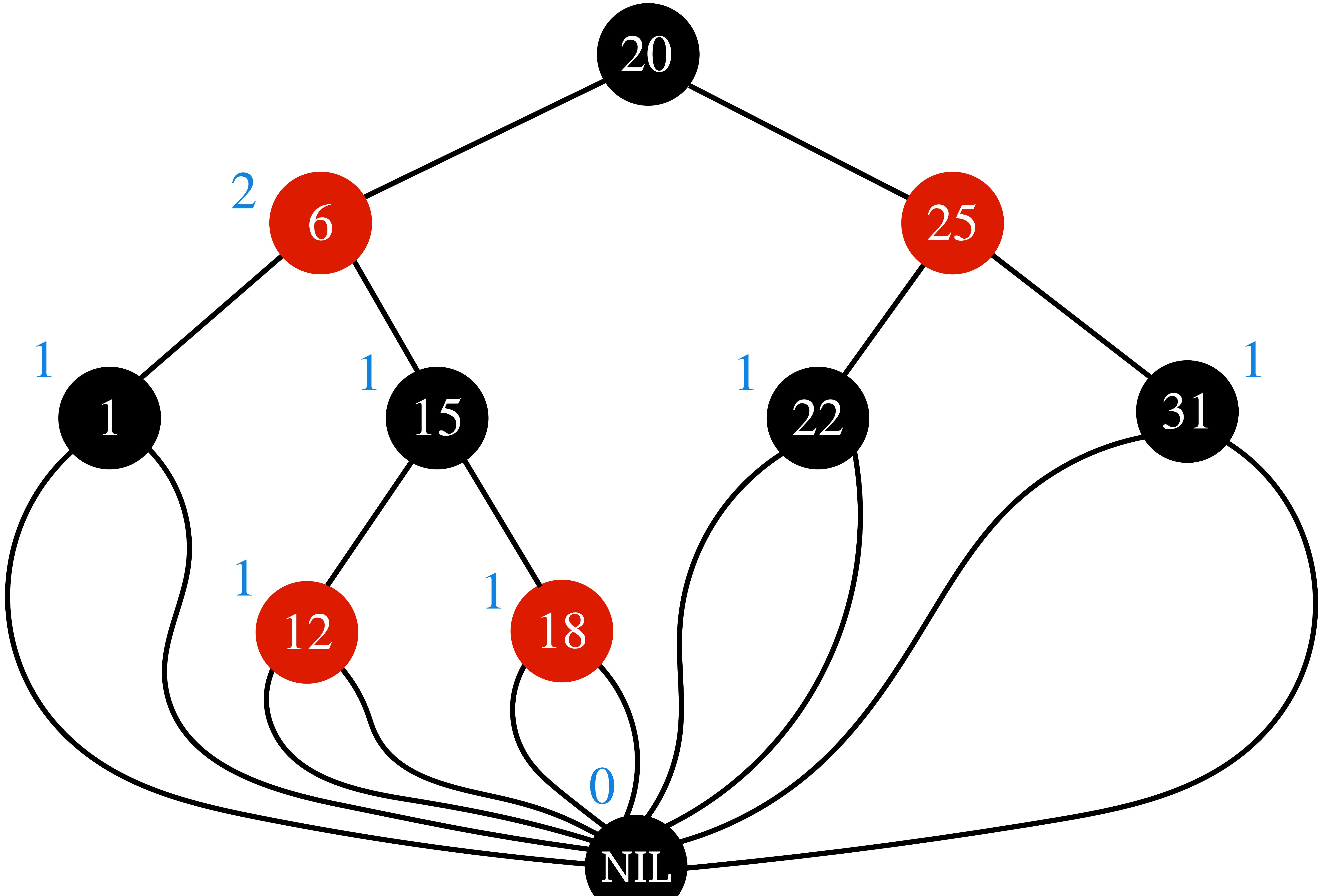
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



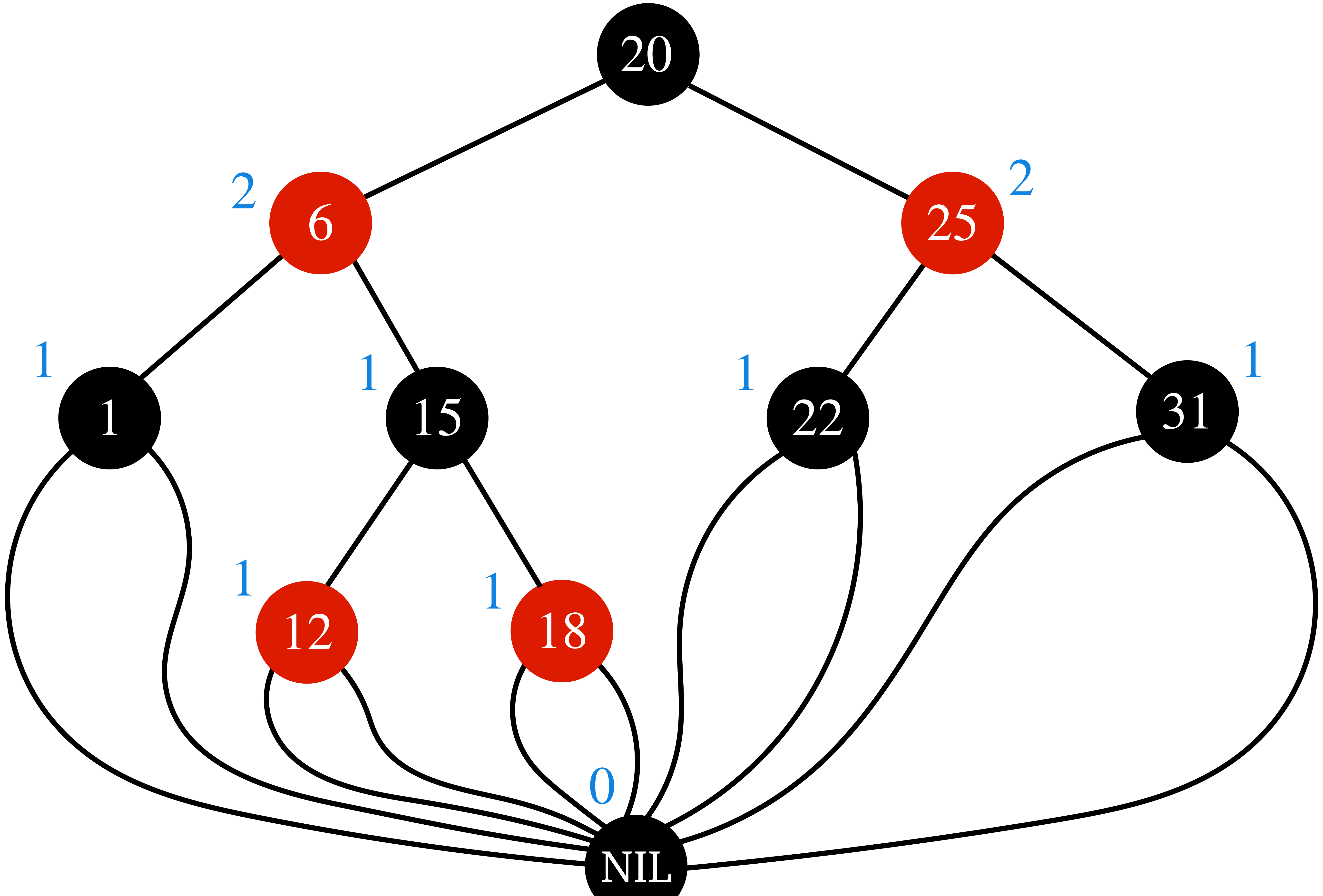
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



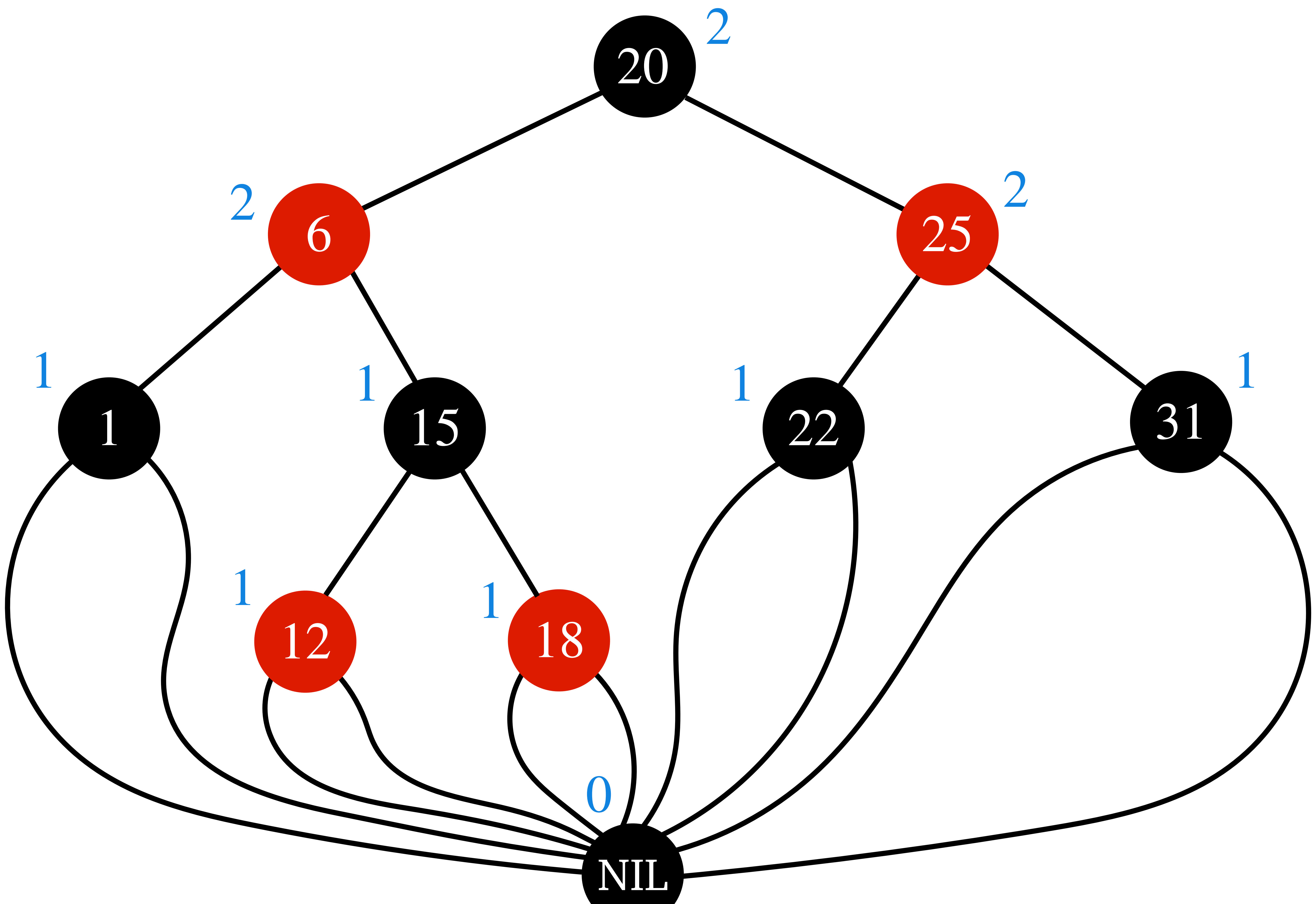
RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



RB-Trees: Black Height

Def: For any node x, the number of black nodes on any path from x to a **leaf**, excluding x, is called **black height** of x, denoted by $bh(x)$.



RB-Trees: Height Bound

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x,

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

RB-Trees: Height Bound

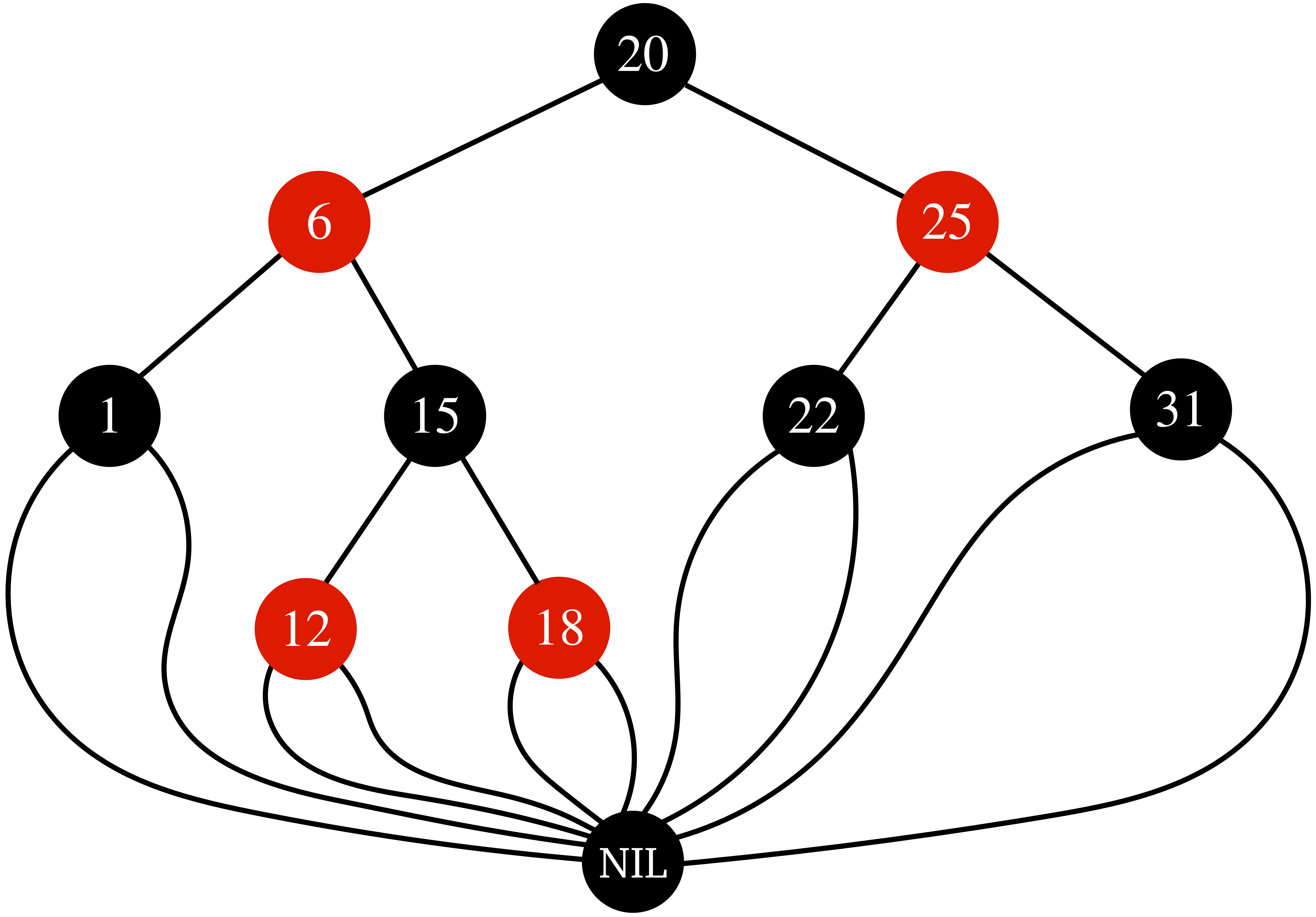
Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Non-NIL nodes

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

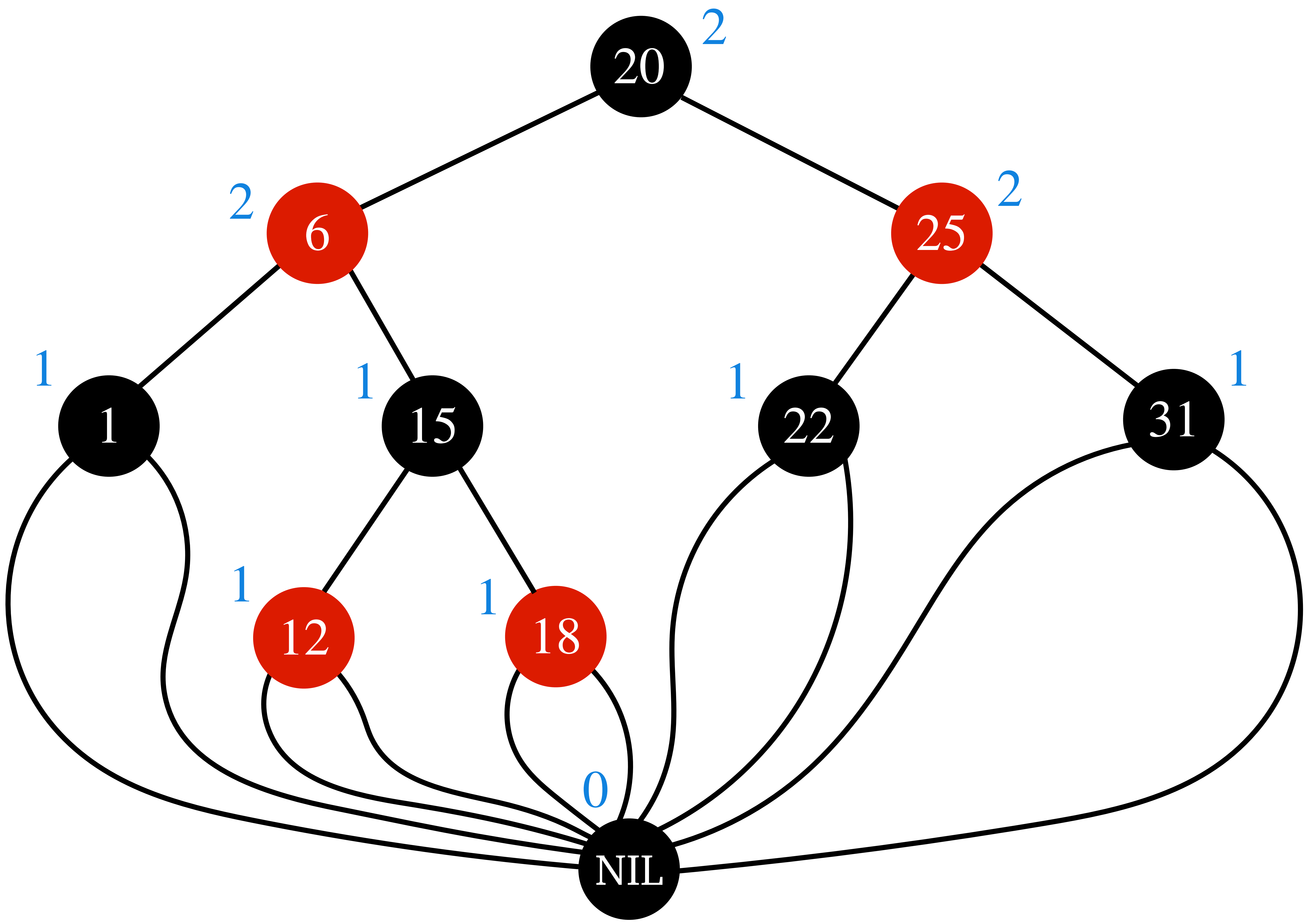
Non-NIL nodes



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Non-NIL nodes



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof:

...

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: Will do induction on the height of the nodes.

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: Will do induction on the height of the nodes.

Basis Step:

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: Will do induction on the height of the nodes.

Basis Step: For nodes of height 0, i.e., NIL node, the claim is trivially true as:

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: Will do induction on the height of the nodes.

Basis Step: For nodes of height 0, i.e., NIL node, the claim is trivially true as:

- $bh(x) = 0$

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: Will do induction on the height of the nodes.

Basis Step: For nodes of height 0, i.e., NIL node, the claim is trivially true as:

- $bh(x) = 0$
- Subtree at NIL node contains $0 = 2^0 - 1$ internal nodes.

…

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof:

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:**

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

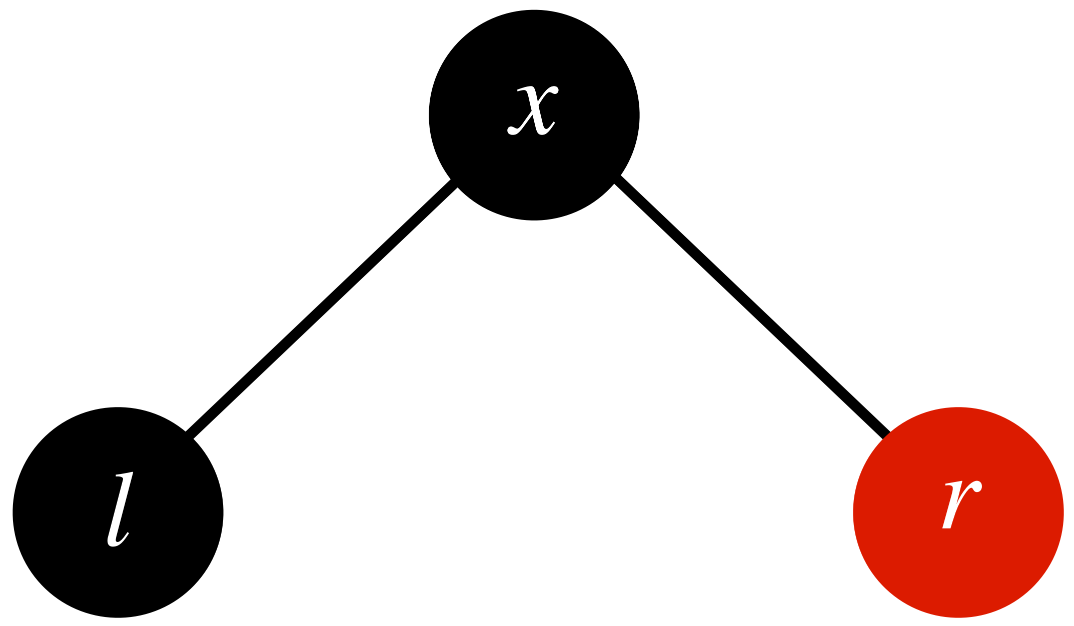
Let x be a node of height $i + 1$, and l and r be its children.

RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.



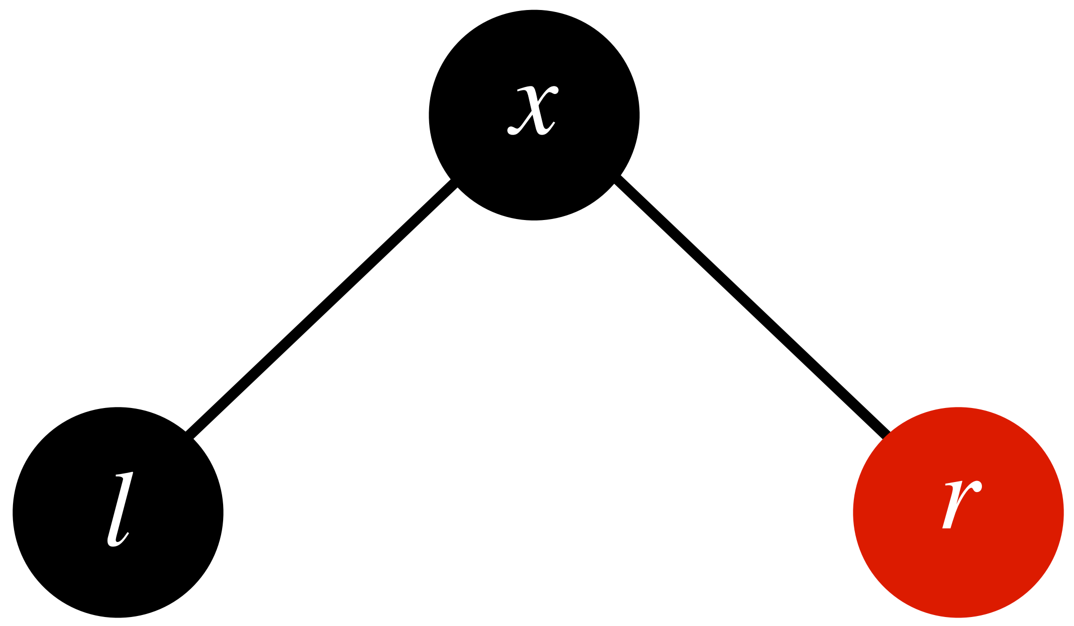
RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)



RB-Trees: Height Bound

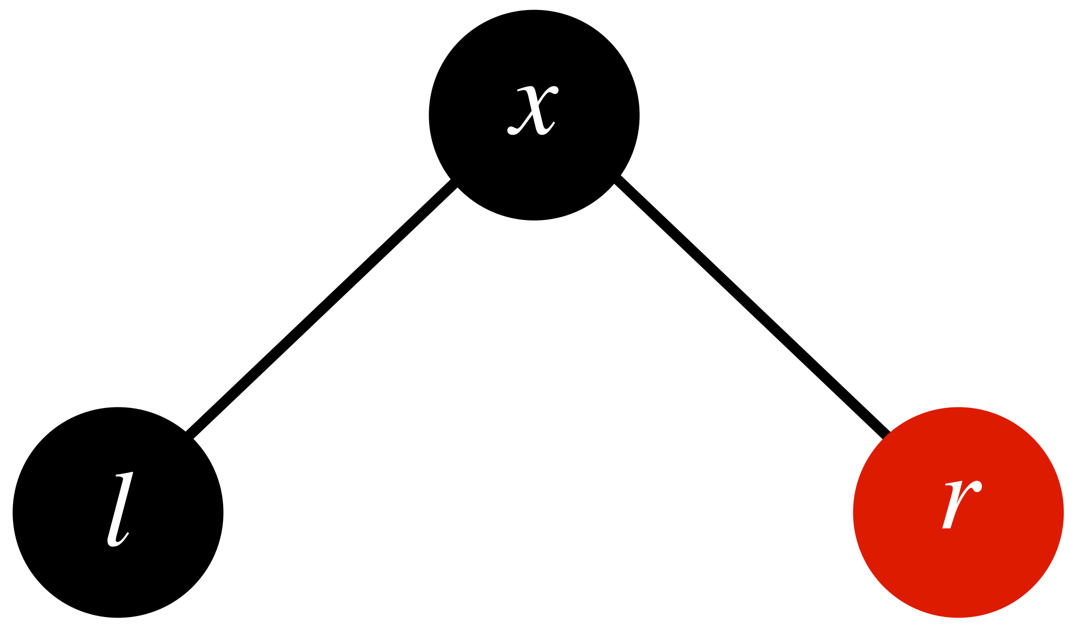
Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)



RB-Trees: Height Bound

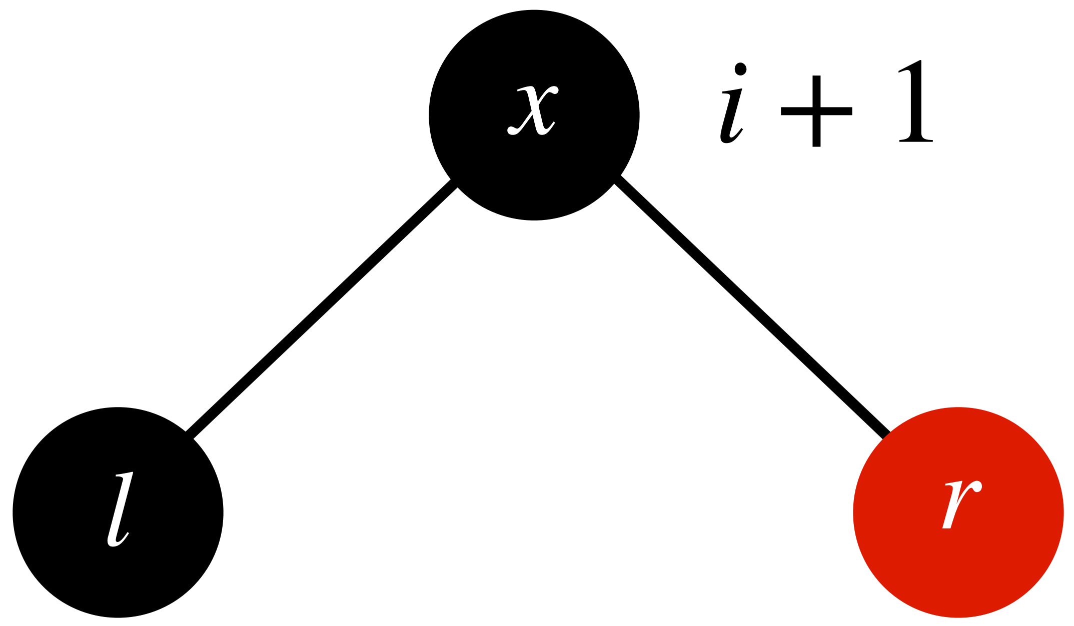
Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)



RB-Trees: Height Bound

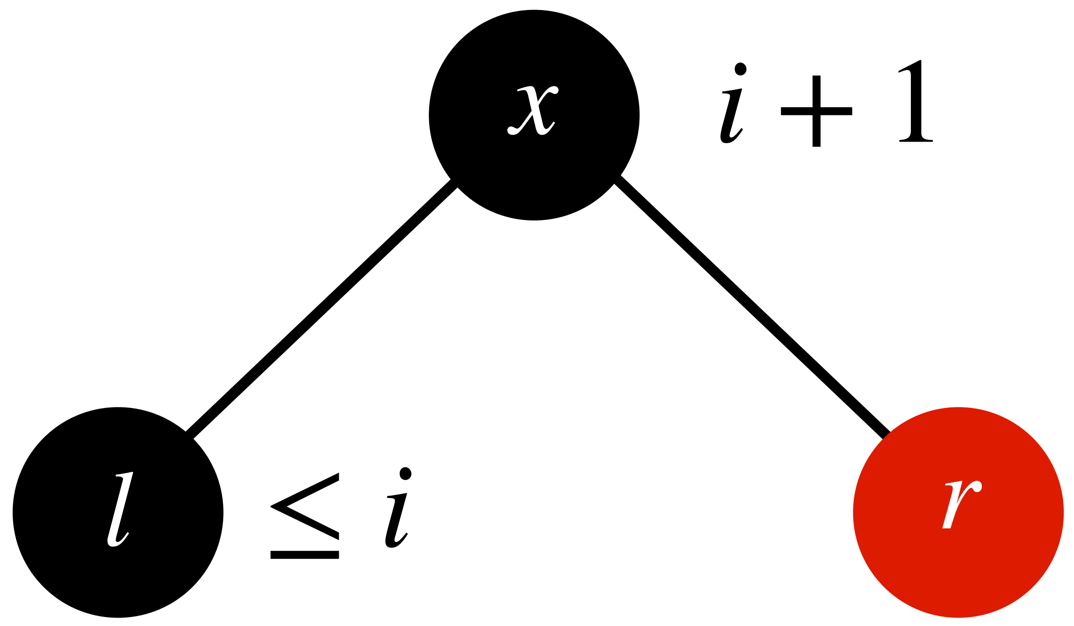
Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)



RB-Trees: Height Bound

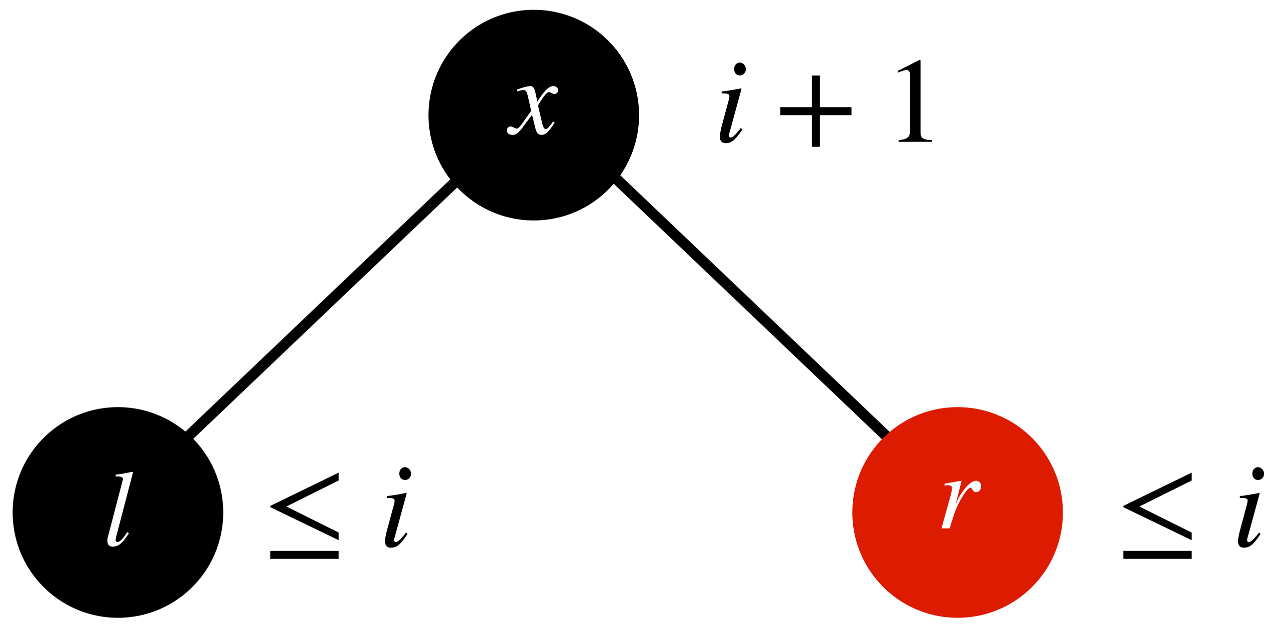
Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

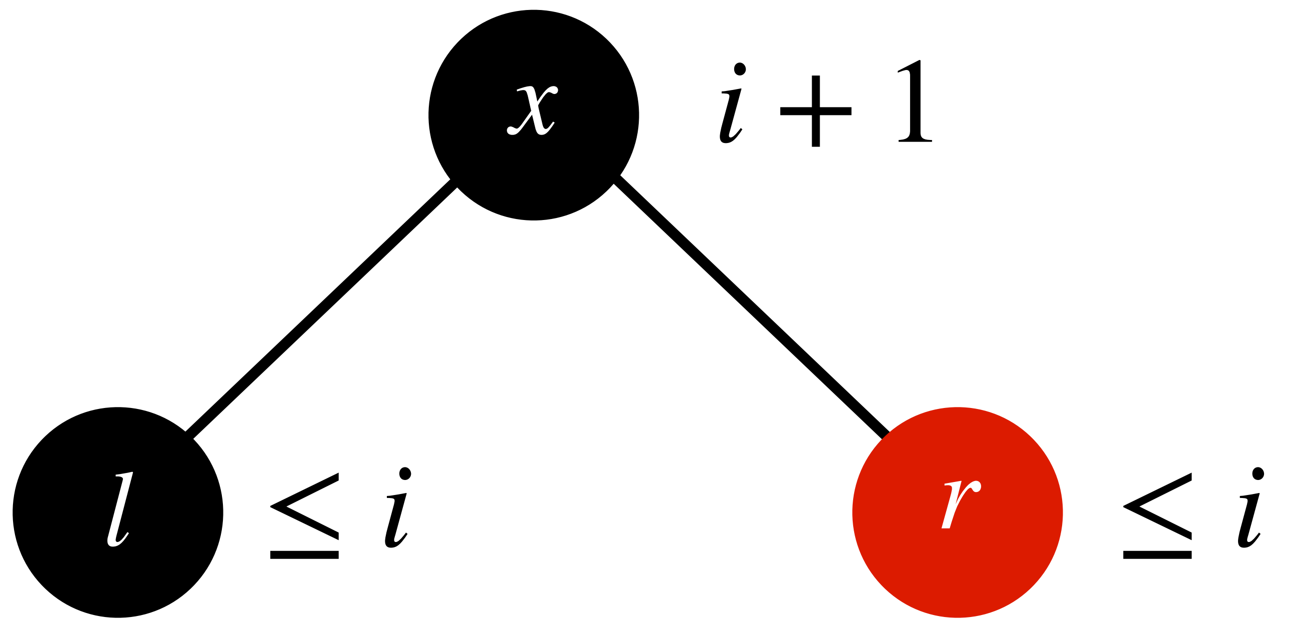
Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

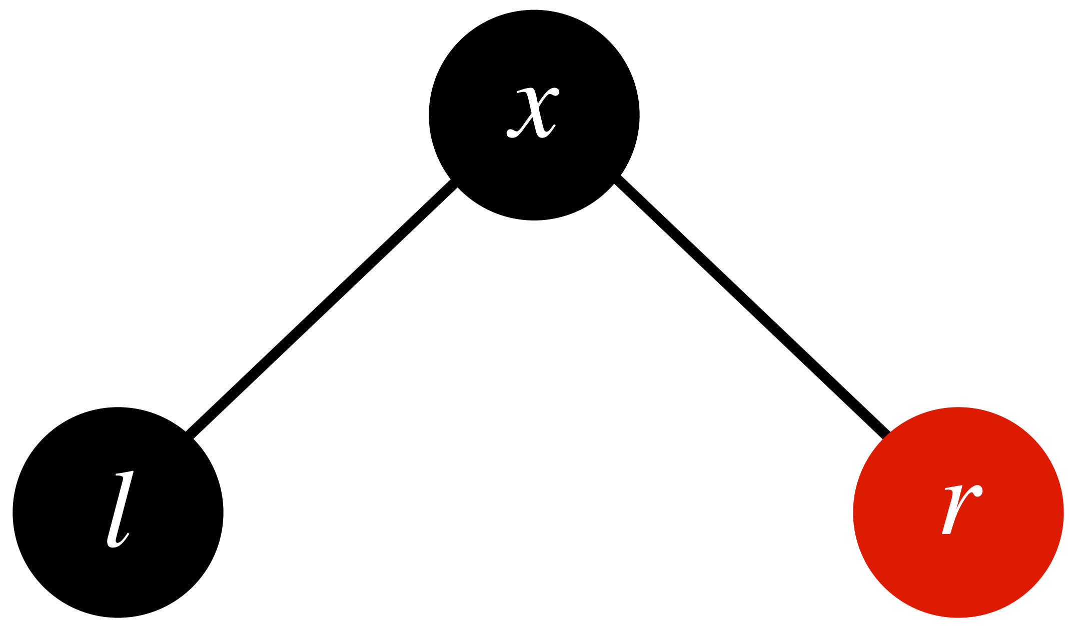
Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

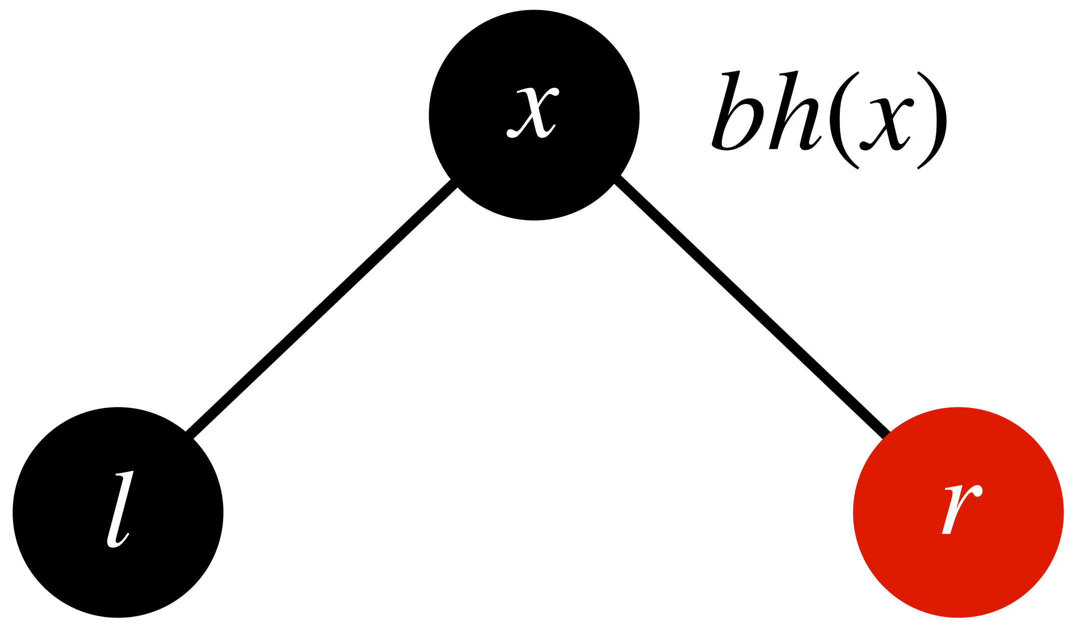
Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

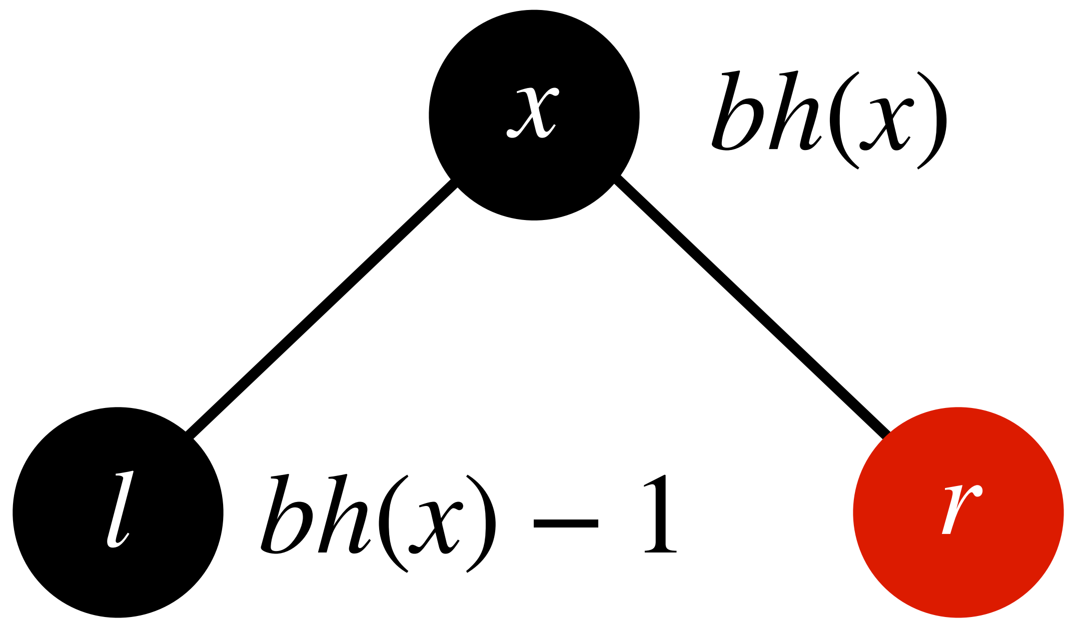
Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

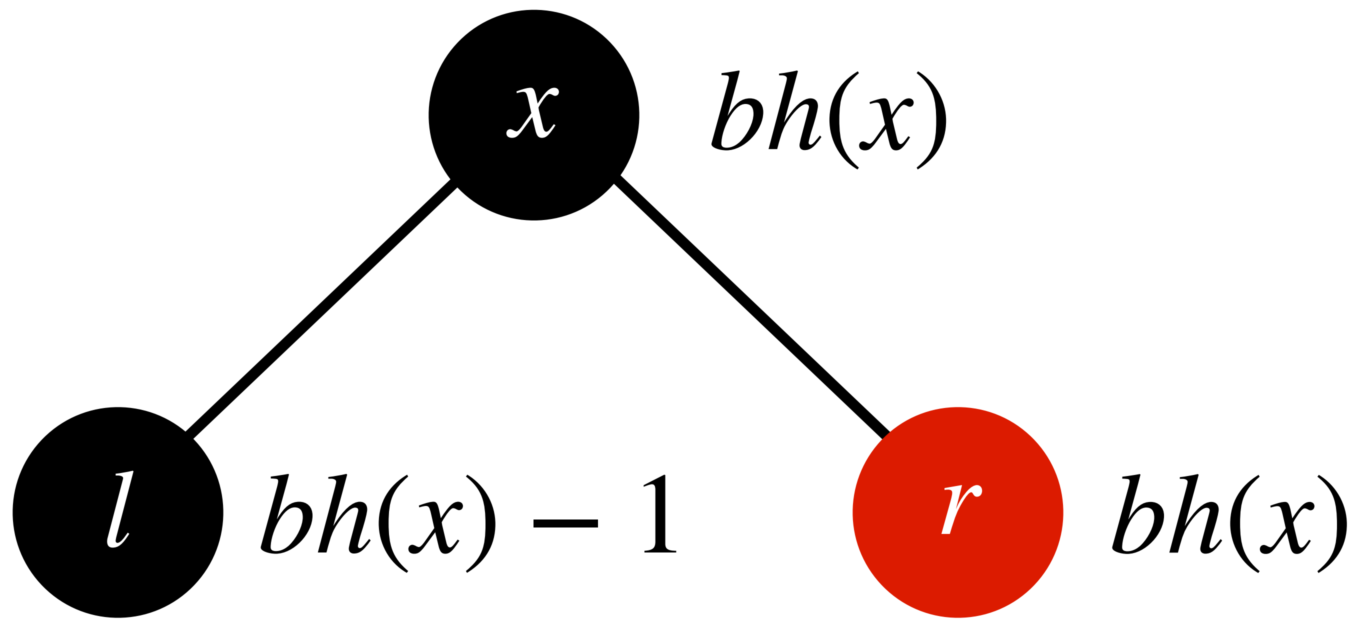
Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

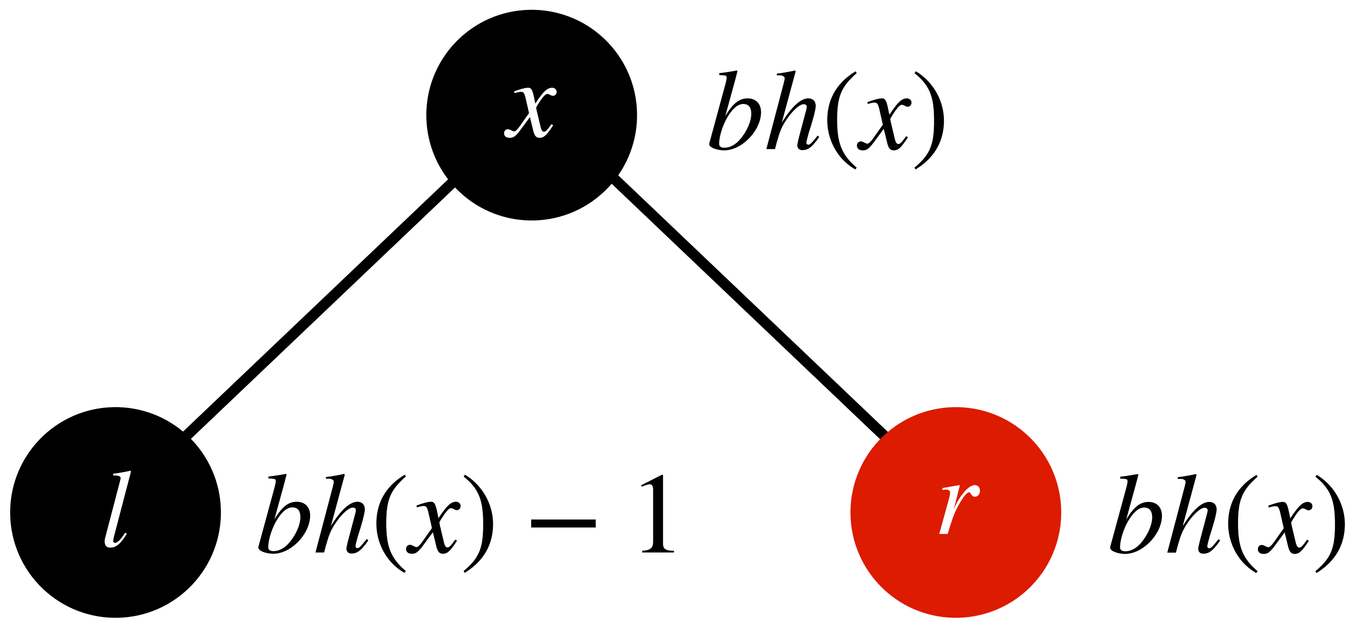
Let x be a node of height $i + 1$, and l and r be its children.

internal nodes in subtree(x)

$= 1 +$ # internal nodes in subtree(l), subtree(r)

$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$

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



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

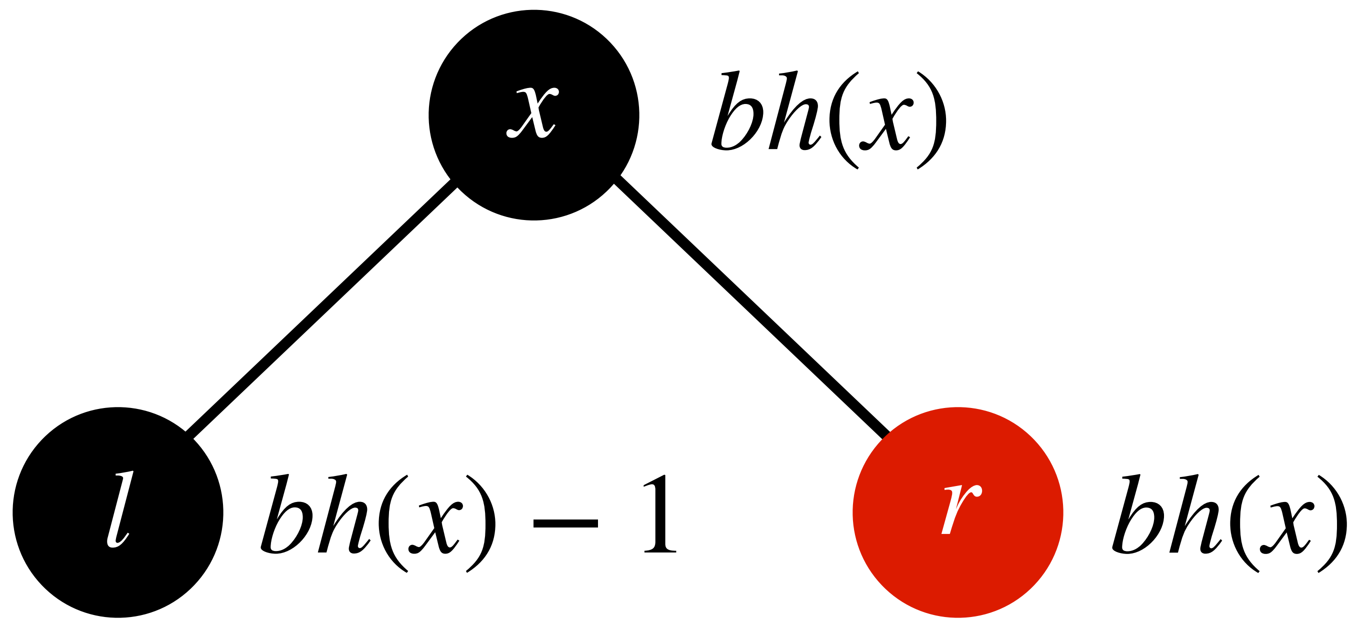
internal nodes in subtree(x)

$$= 1 + \text{\# internal nodes in subtree}(l), \text{subtree}(r)$$

$$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$$

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

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



RB-Trees: Height Bound

Claim: In any RB-tree, every subtree rooted at any node, say x, contains at least $2^{bh(x)} - 1$ internal nodes.

Proof: **Inductive Step:** Assume the claim is true for any node of height $\leq i$.

Let x be a node of height $i + 1$, and l and r be its children.

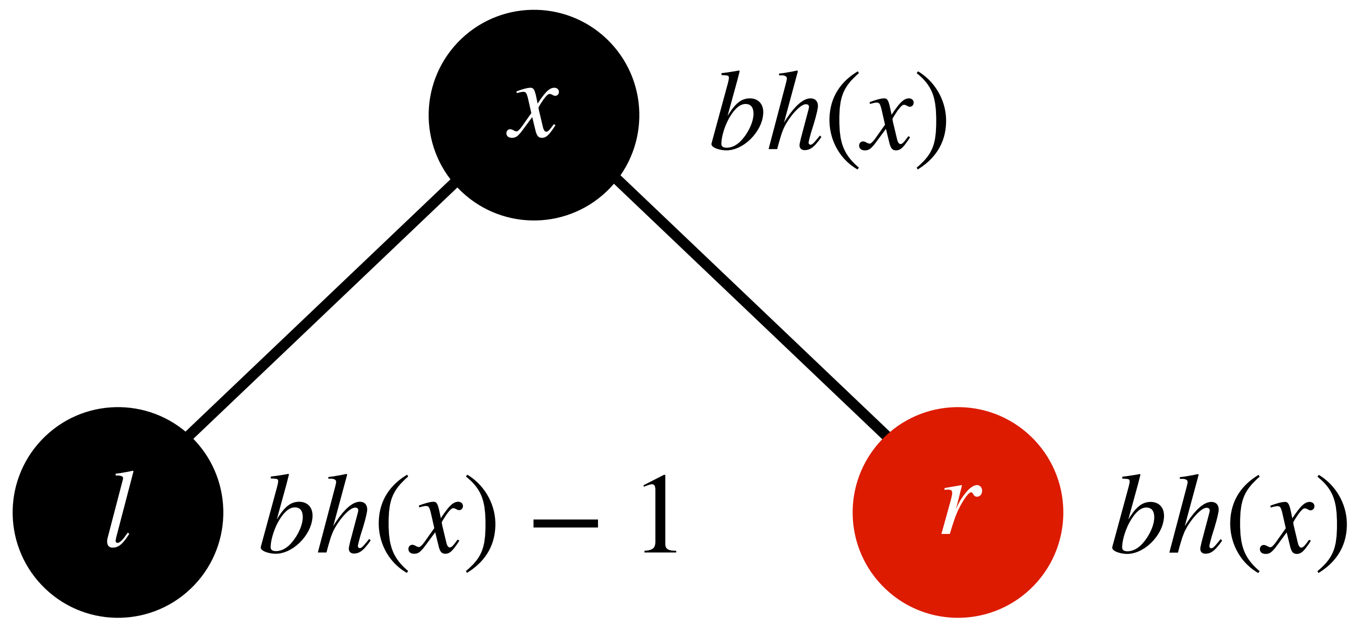
internal nodes in subtree(x)

$$= 1 + \text{\# internal nodes in subtree}(l), \text{subtree}(r)$$

$$\geq 1 + 2^{bh(l)} - 1 + 2^{bh(r)} - 1$$

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

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



∎

RB-Trees: Height Bound

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof:

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

RB-Trees: Height Bound

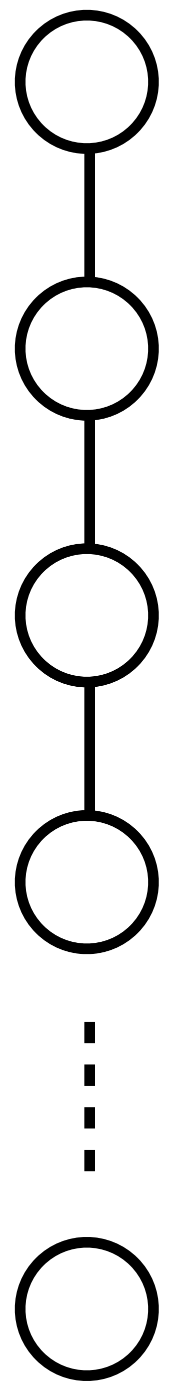
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

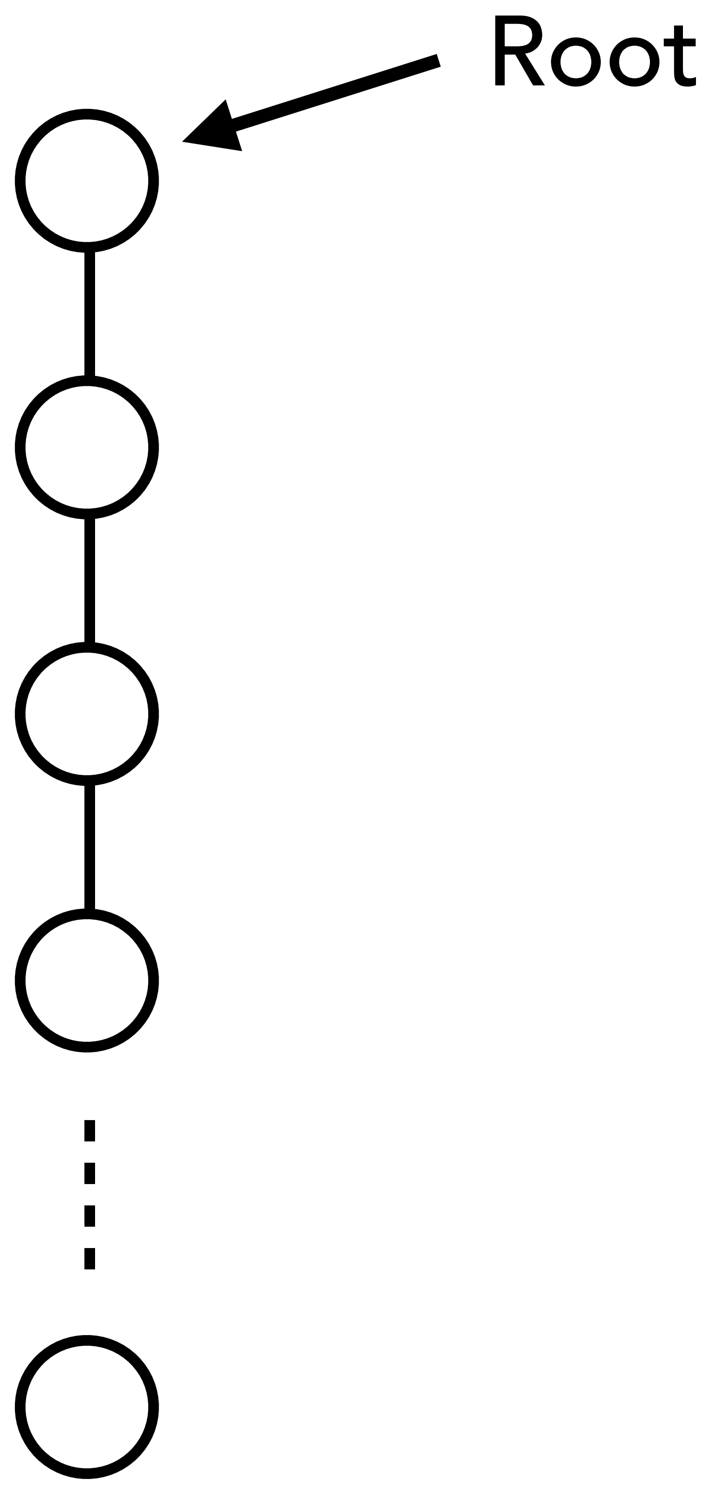
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

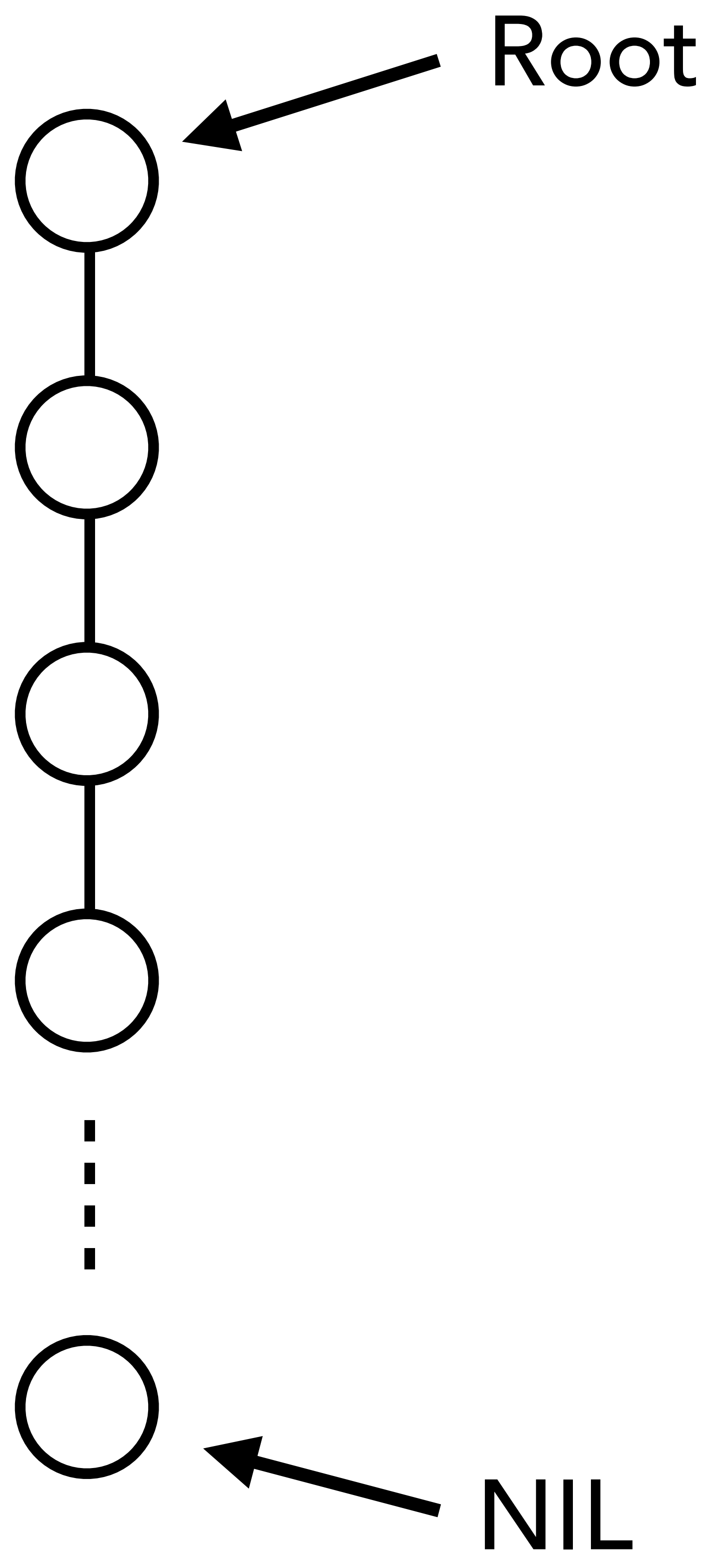
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

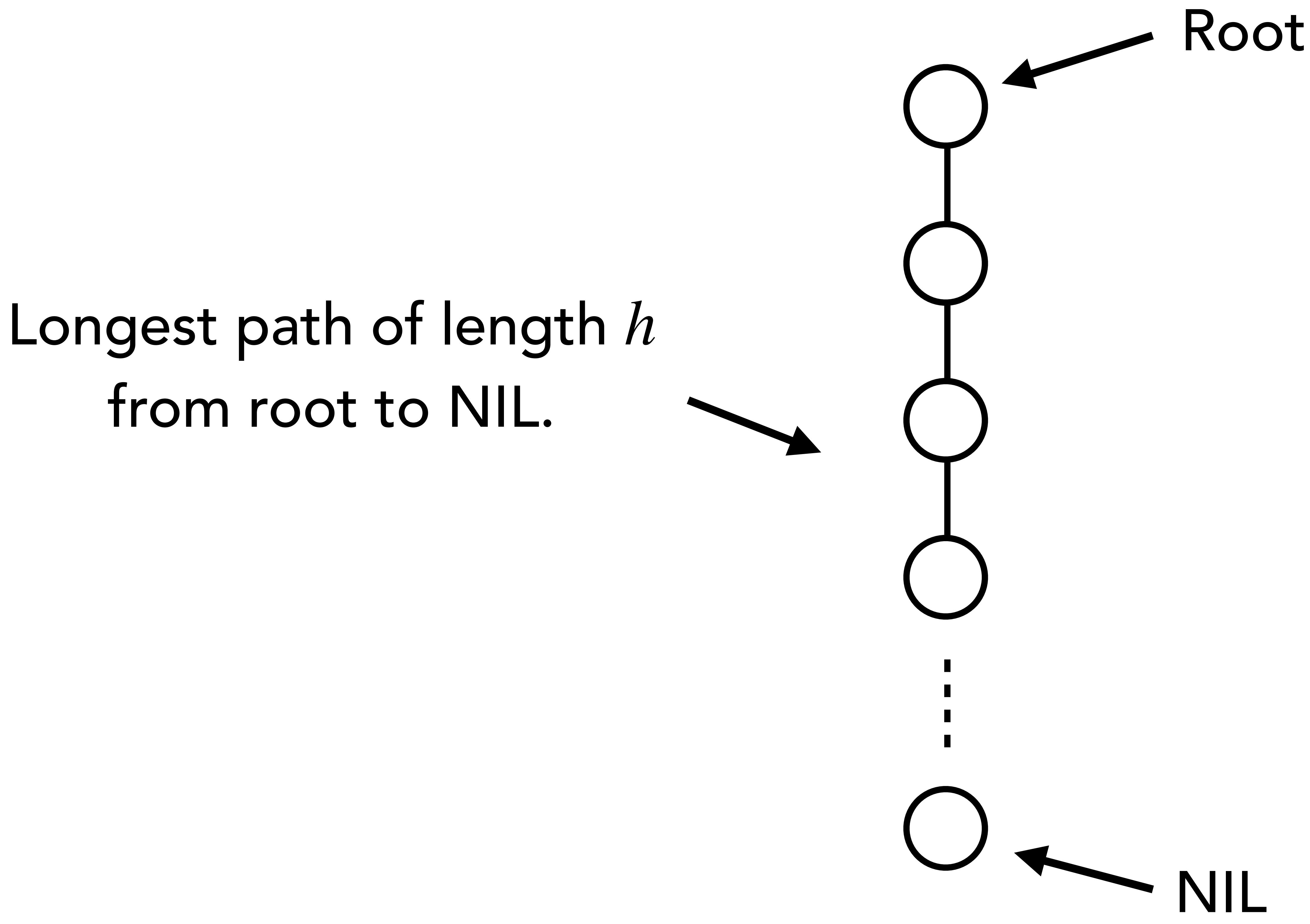
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

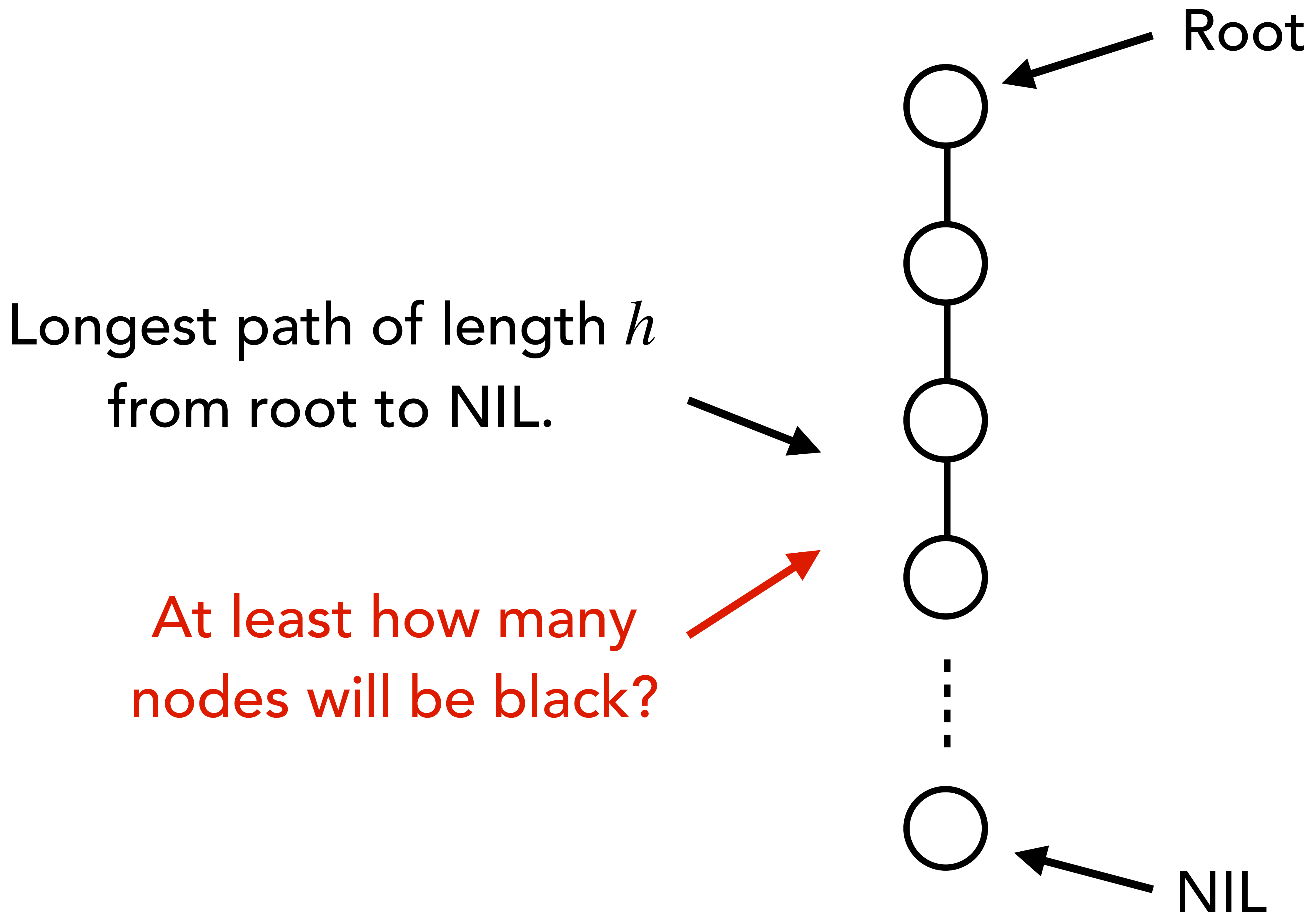
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

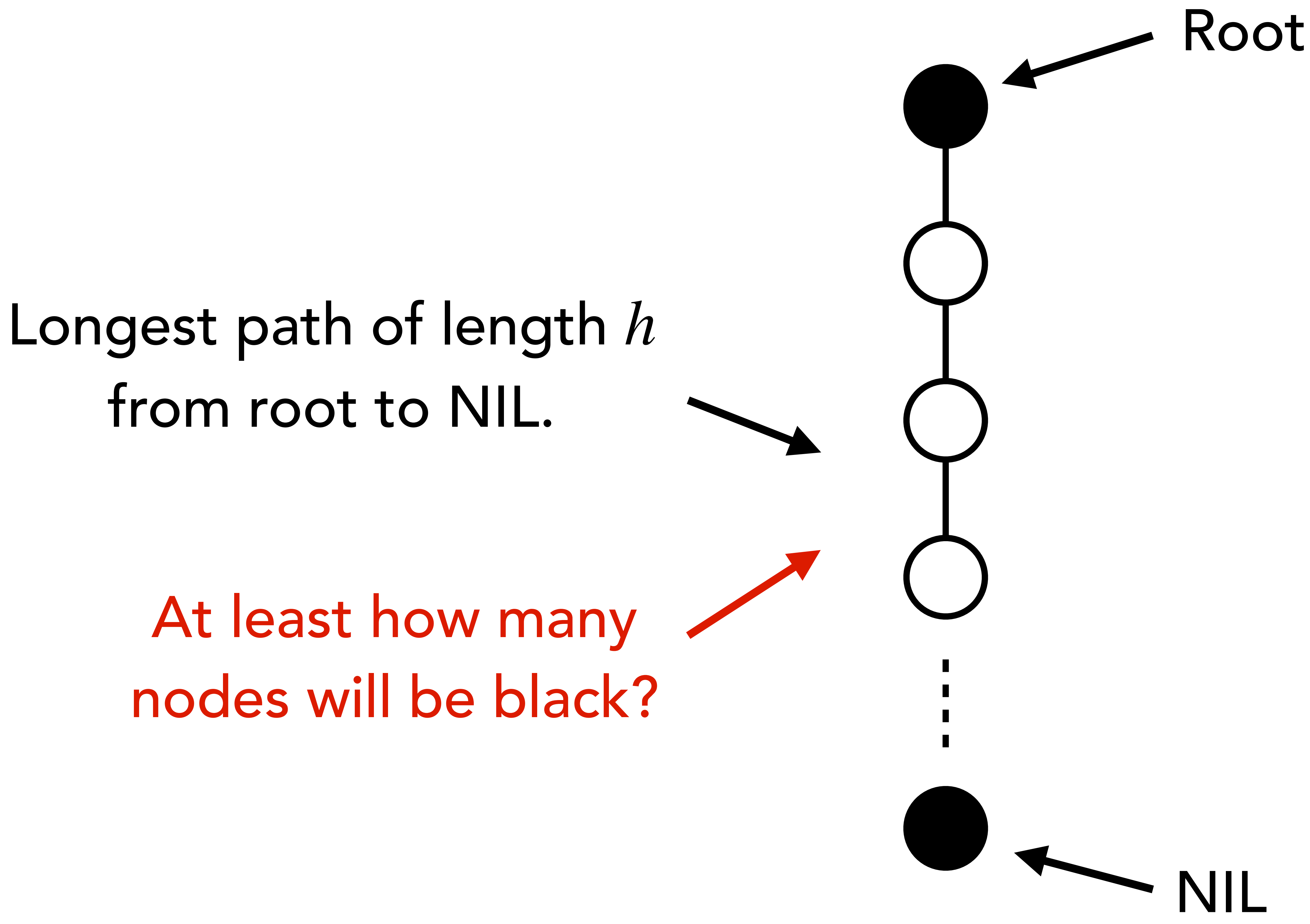
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

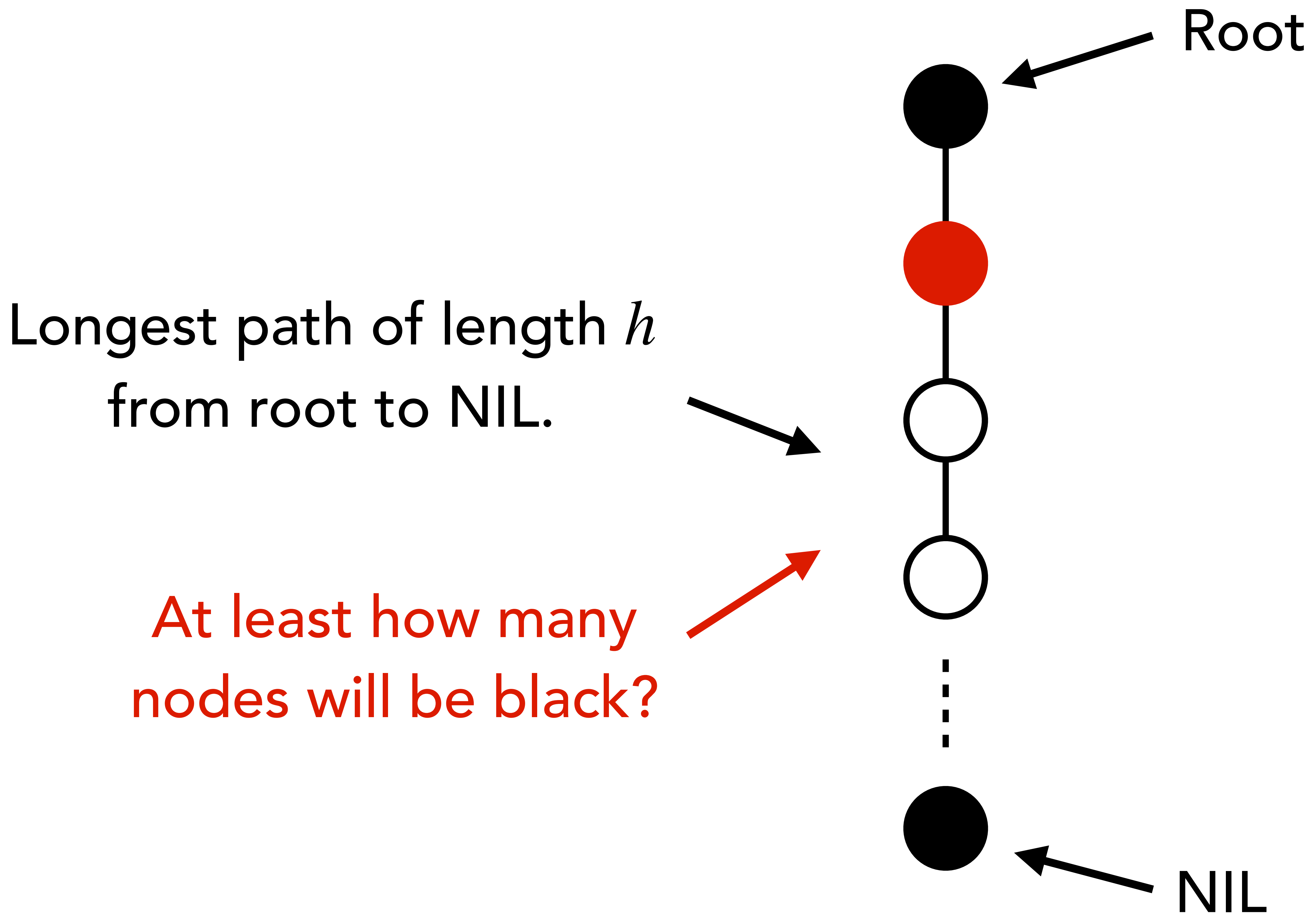
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

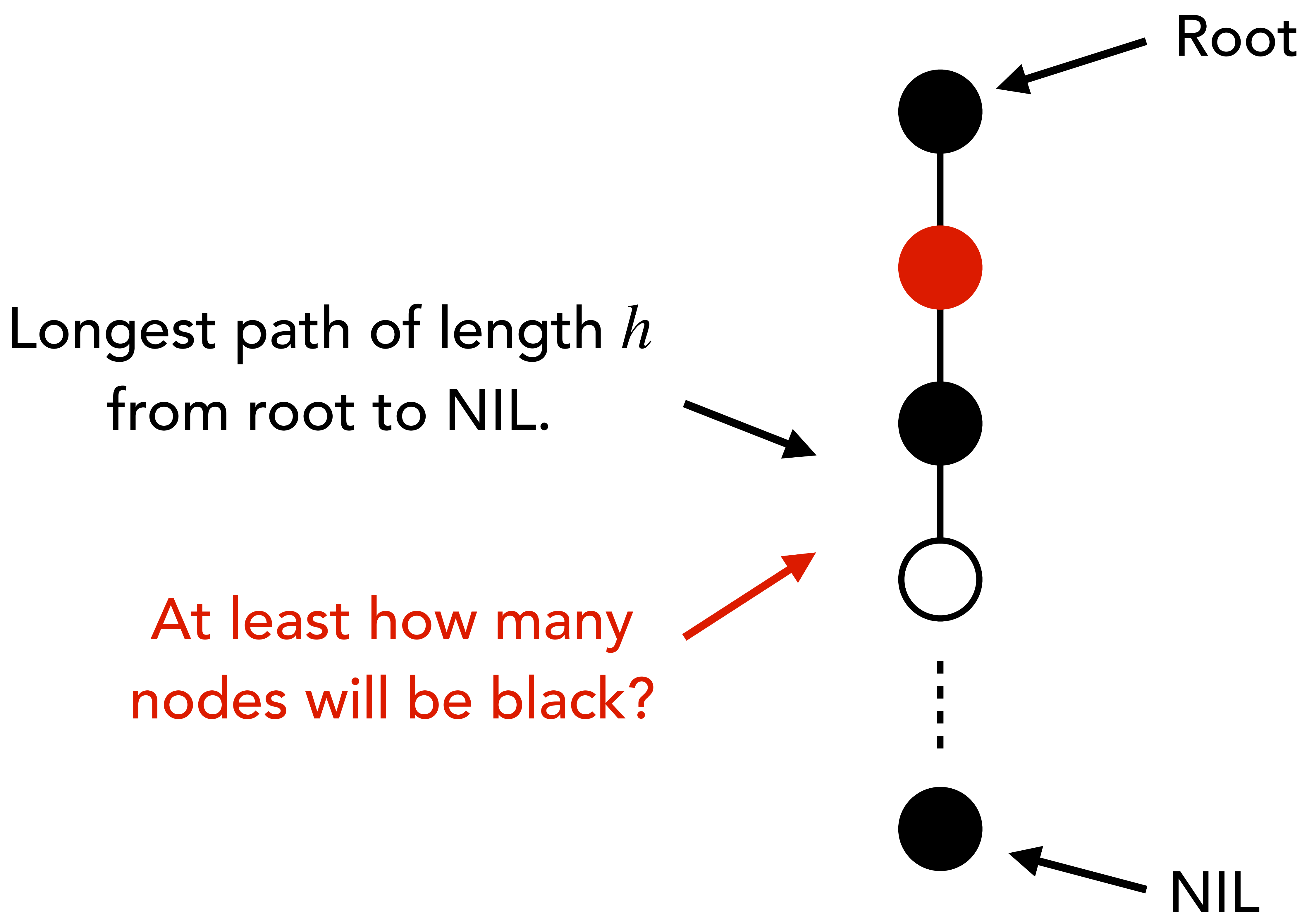
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

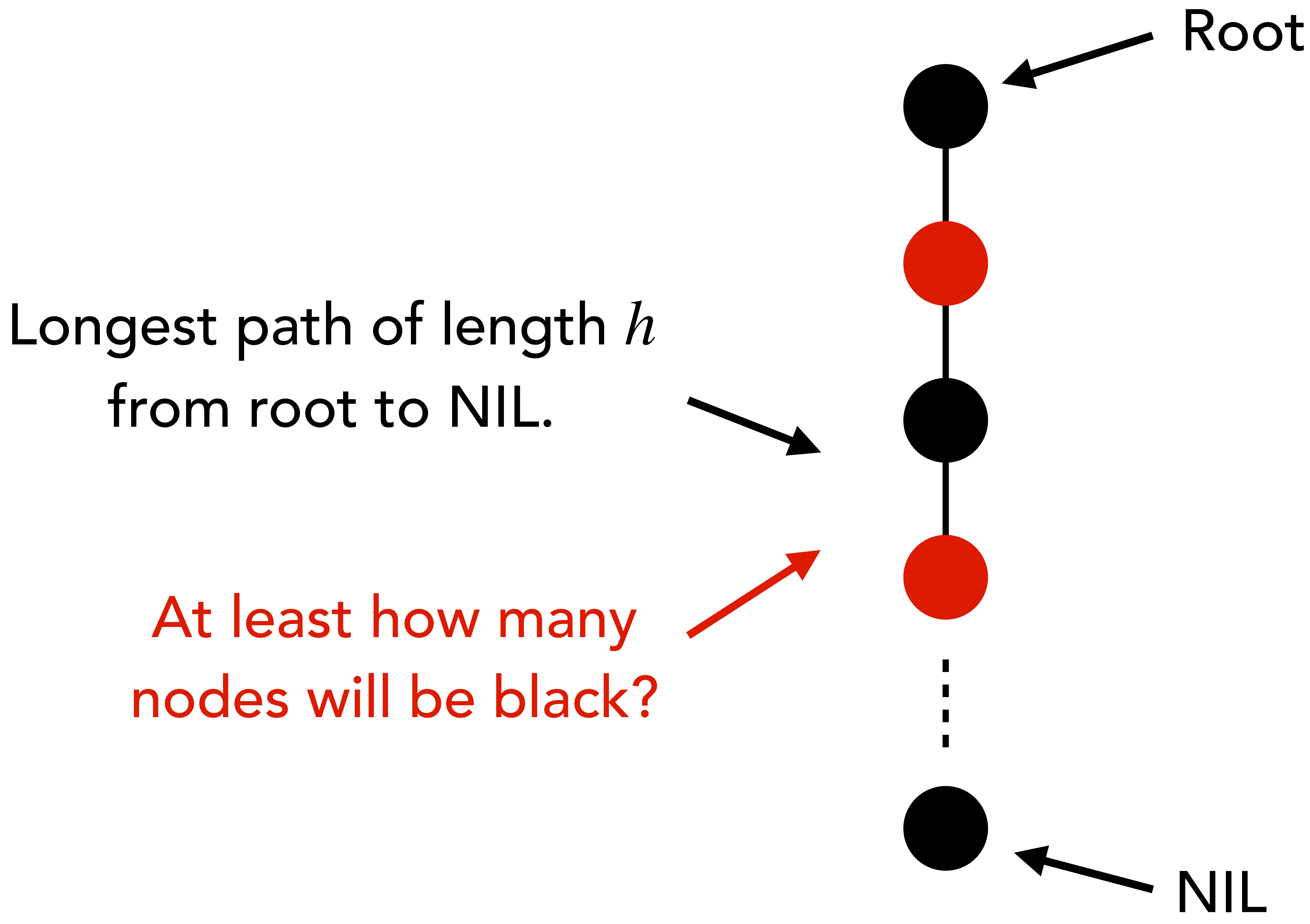
Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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



RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\Longrightarrow \quad n \geq 2^{h/2} - 1$$

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\implies \quad n \geq 2^{h/2} - 1 \quad (\because \text{At least half of the nodes must be black on longest path})$$

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\implies n \geq 2^{h/2} - 1 \quad (\because \text{At least half of the nodes must be black on longest path})$$

$$\implies n + 1 \geq 2^{h/2}$$

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\implies n \geq 2^{h/2} - 1 \quad (\because \text{At least half of the nodes must be black on longest path})$$

$$\implies n + 1 \geq 2^{h/2}$$

$$\implies \lg(n+1) \geq h/2$$

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\implies n \geq 2^{h/2} - 1 \quad (\because \text{At least half of the nodes must be black on longest path})$$

$$\implies n + 1 \geq 2^{h/2}$$

$$\implies \lg(n+1) \geq h/2$$

$$\implies h \leq 2\lg(n+1)$$

RB-Trees: Height Bound

Lemma: A red-black tree with n internal nodes has height at most $2\lg(n+1)$.

Proof: Let h be the height of root and the tree.

We know that subtree at root contains least $2^{bh(root)} - 1$ internal nodes.

Therefore,

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

$$\implies n \geq 2^{h/2} - 1 \quad (\because \text{At least half of the nodes must be black on longest path})$$

$$\implies n + 1 \geq 2^{h/2}$$

$$\implies \lg(n+1) \geq h/2$$

$$\implies h \leq 2\lg(n+1)$$

■

RB-Trees: Operations

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.
- Search for an element with the key k.

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.
- Search for an element with the key k.
- Minimum or Maximum of the set.

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.
- Search for an element with the key k.
- Minimum or Maximum of the set.
- Successor or Predecessor of an element of the set.

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.
- Search for an element with the key k.
- Minimum or Maximum of the set.
- Successor or Predecessor of an element of the set.

Same as BSTs

RB-Trees: Operations

We can perform the same operations on RB-Trees as we did on BSTs:

- Insert an element.
- Delete an element.
- Search for an element with the key k.
- Minimum or Maximum of the set.
- Successor or Predecessor of an element of the set.

Require care

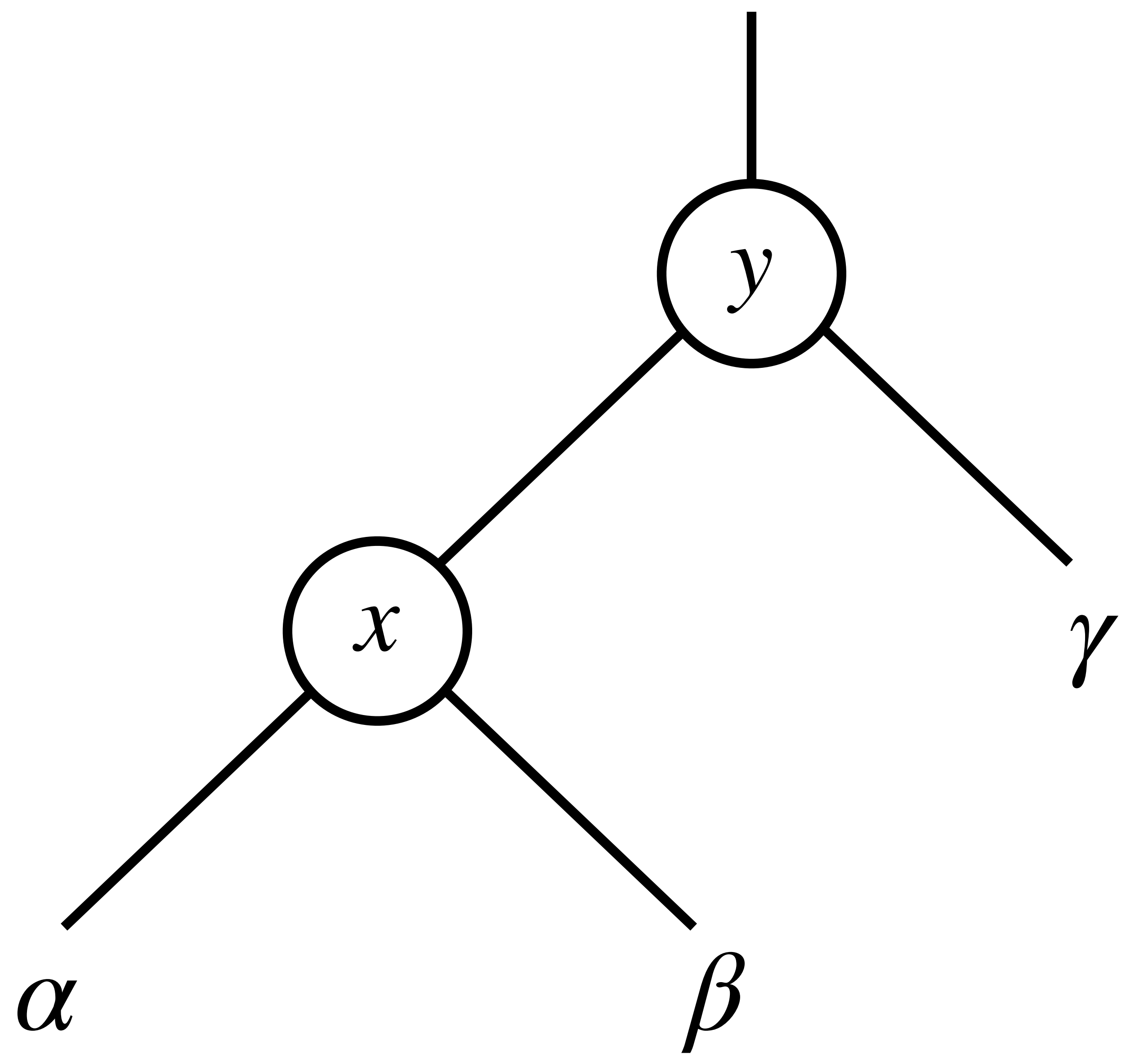
RB-Trees: Rotations

RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:

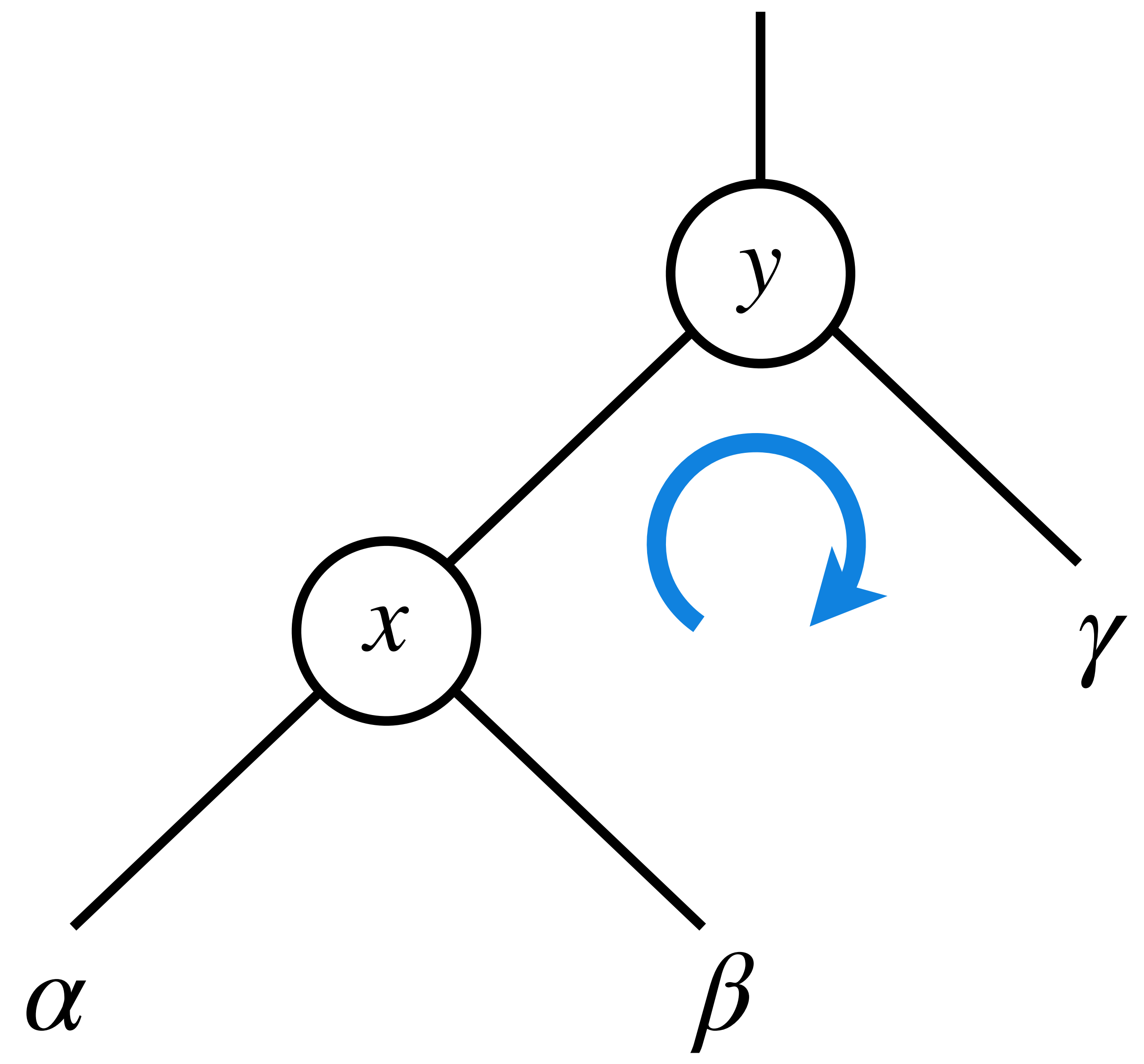
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



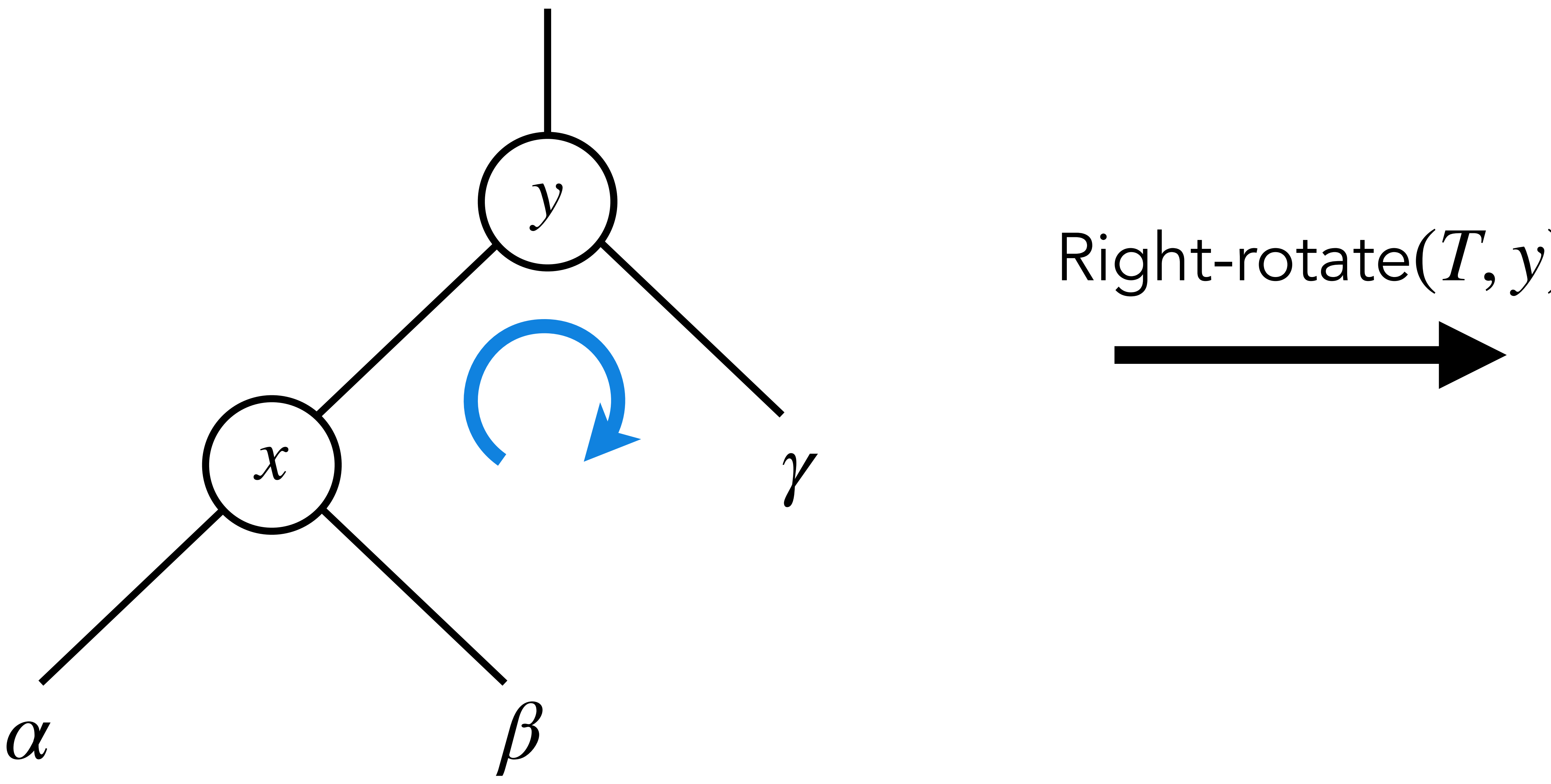
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



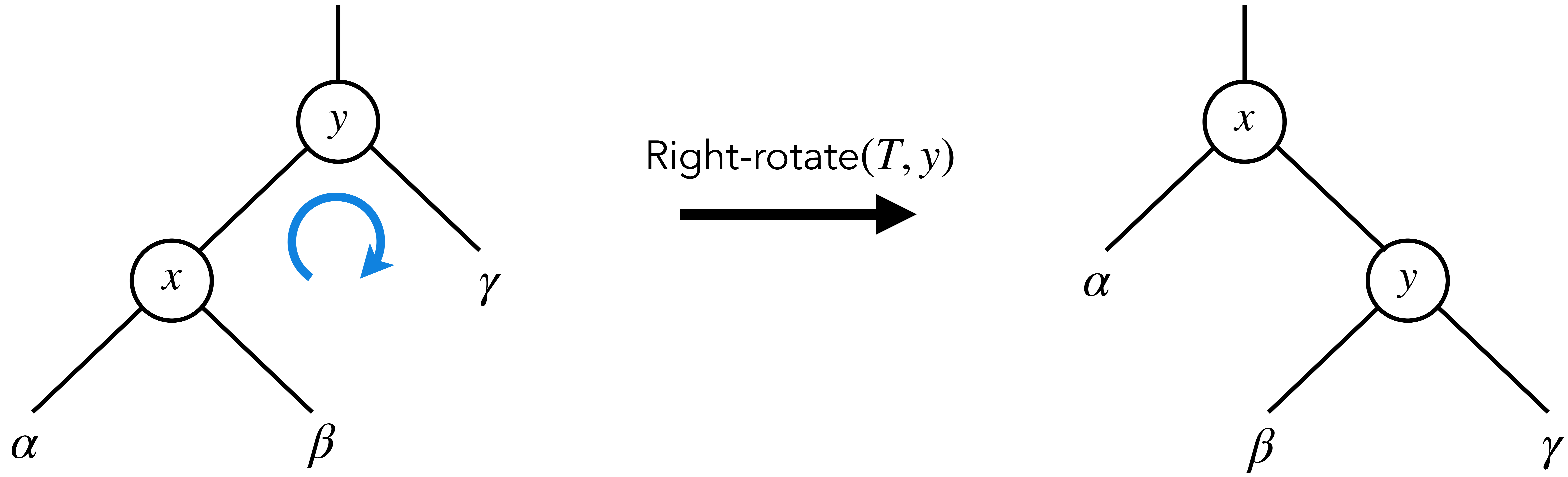
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



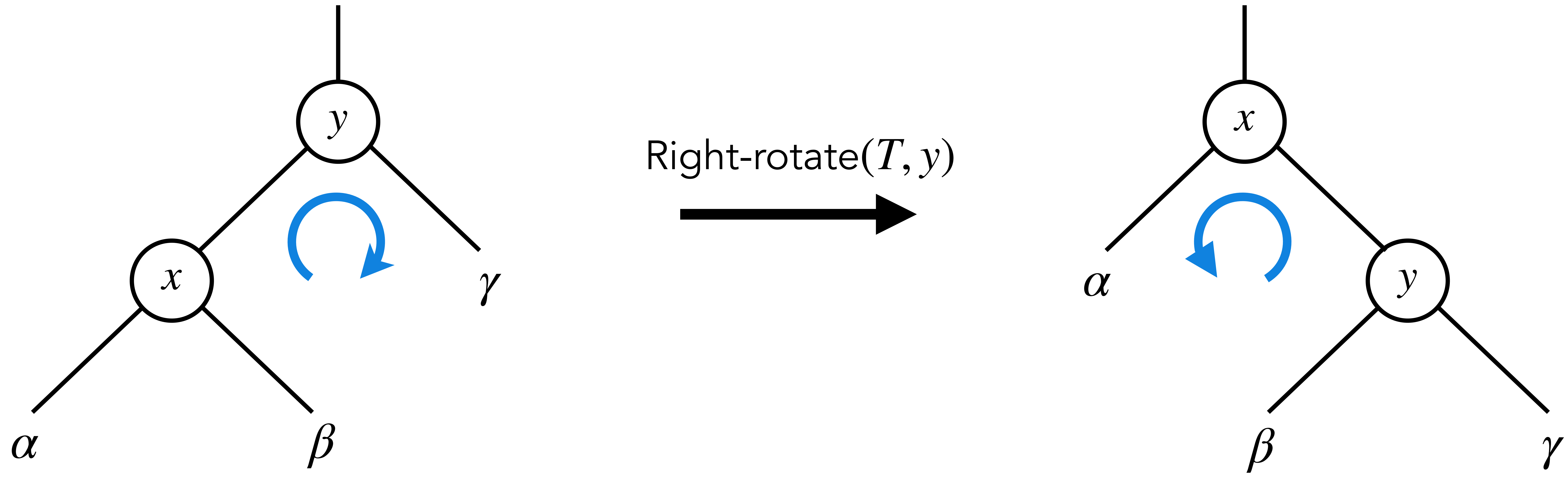
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



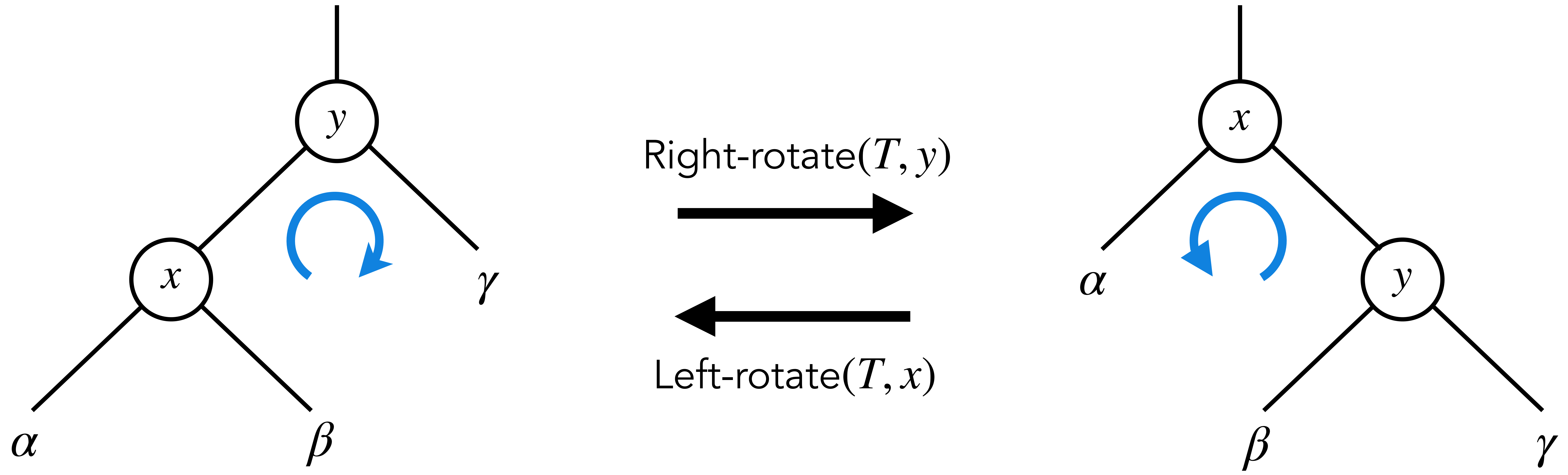
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



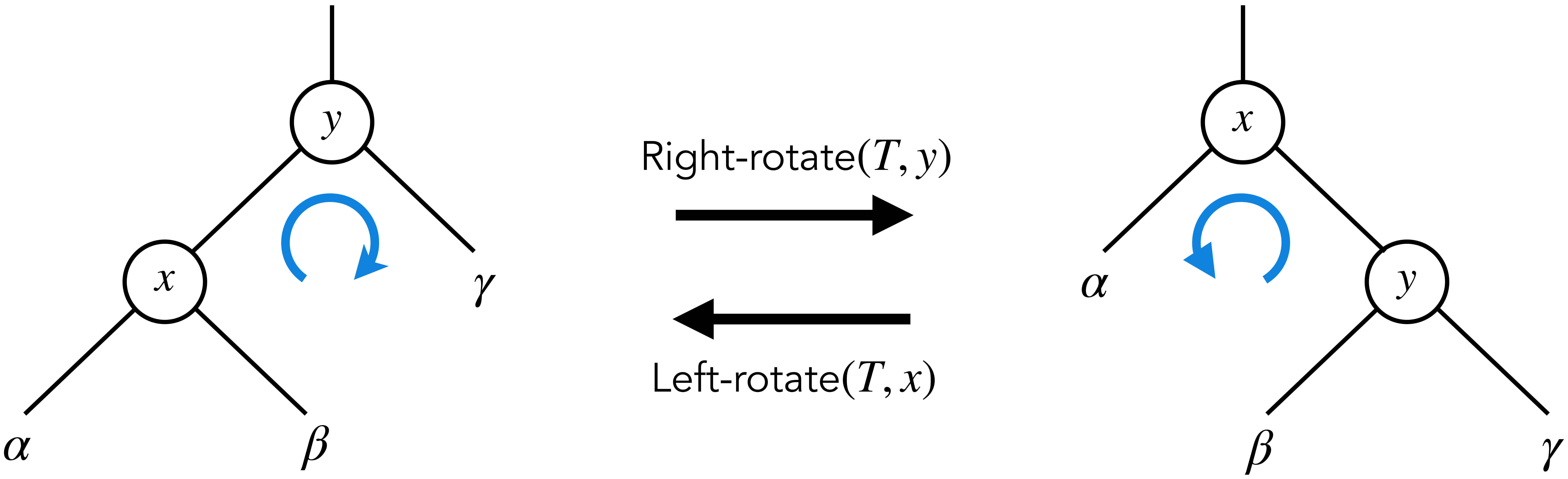
RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



RB-Trees: Rotations

Rotations are basic operations useful in Insertion and Deletion on an RB-tree:



Note: Rotations do not disturb BST property and can be performed in constant time.

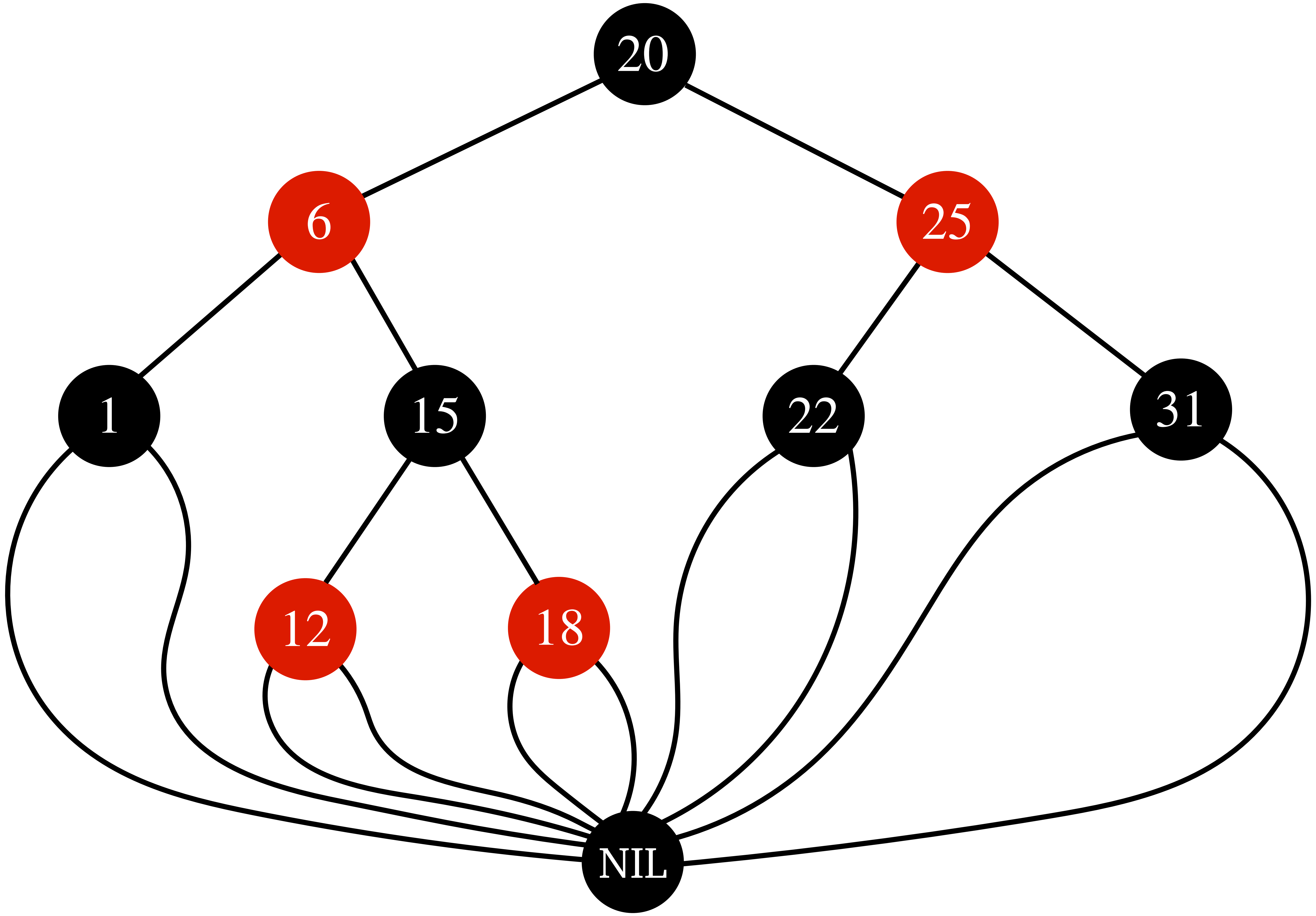
RB-Trees: Insertion

RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:

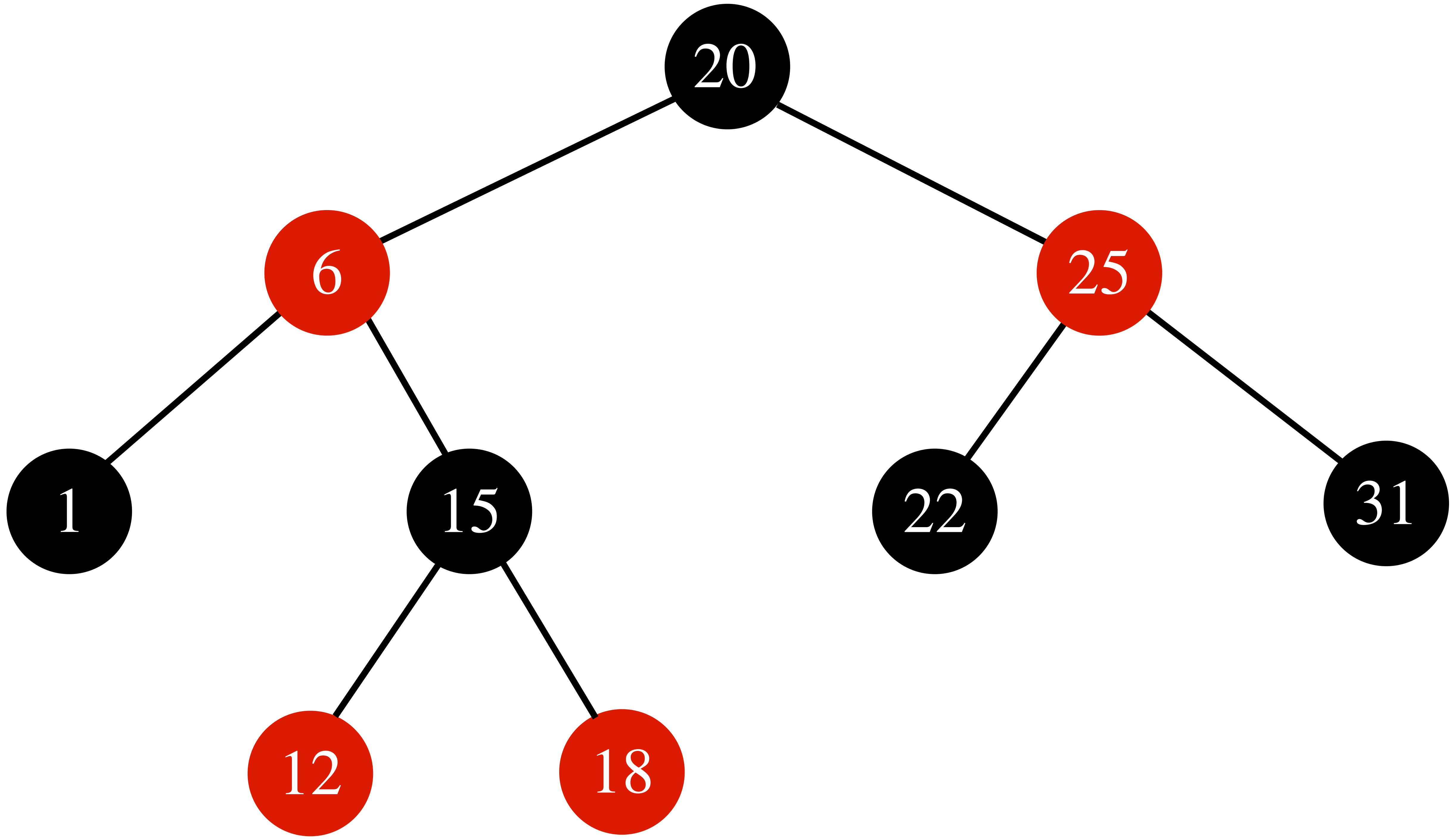
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



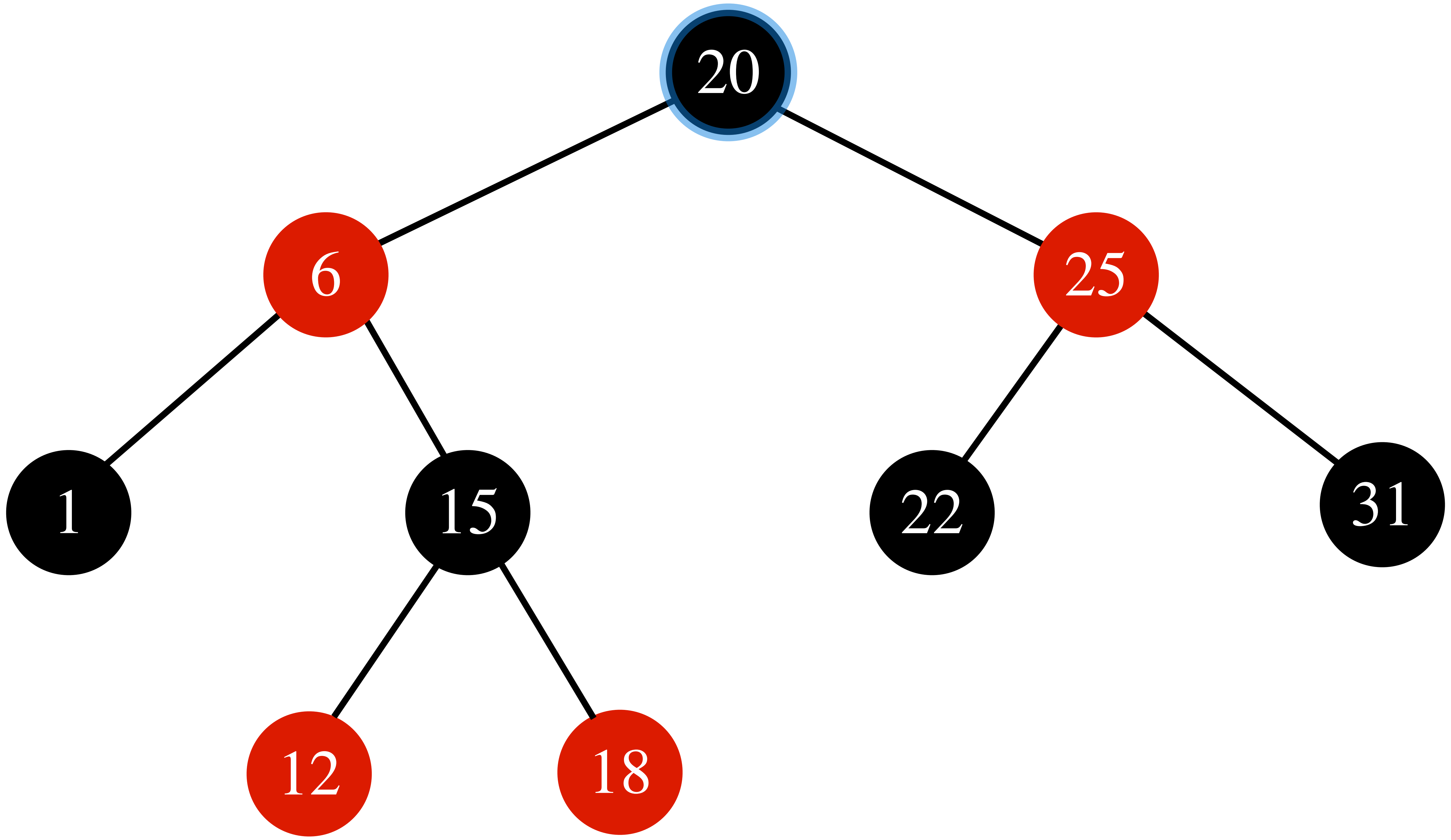
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



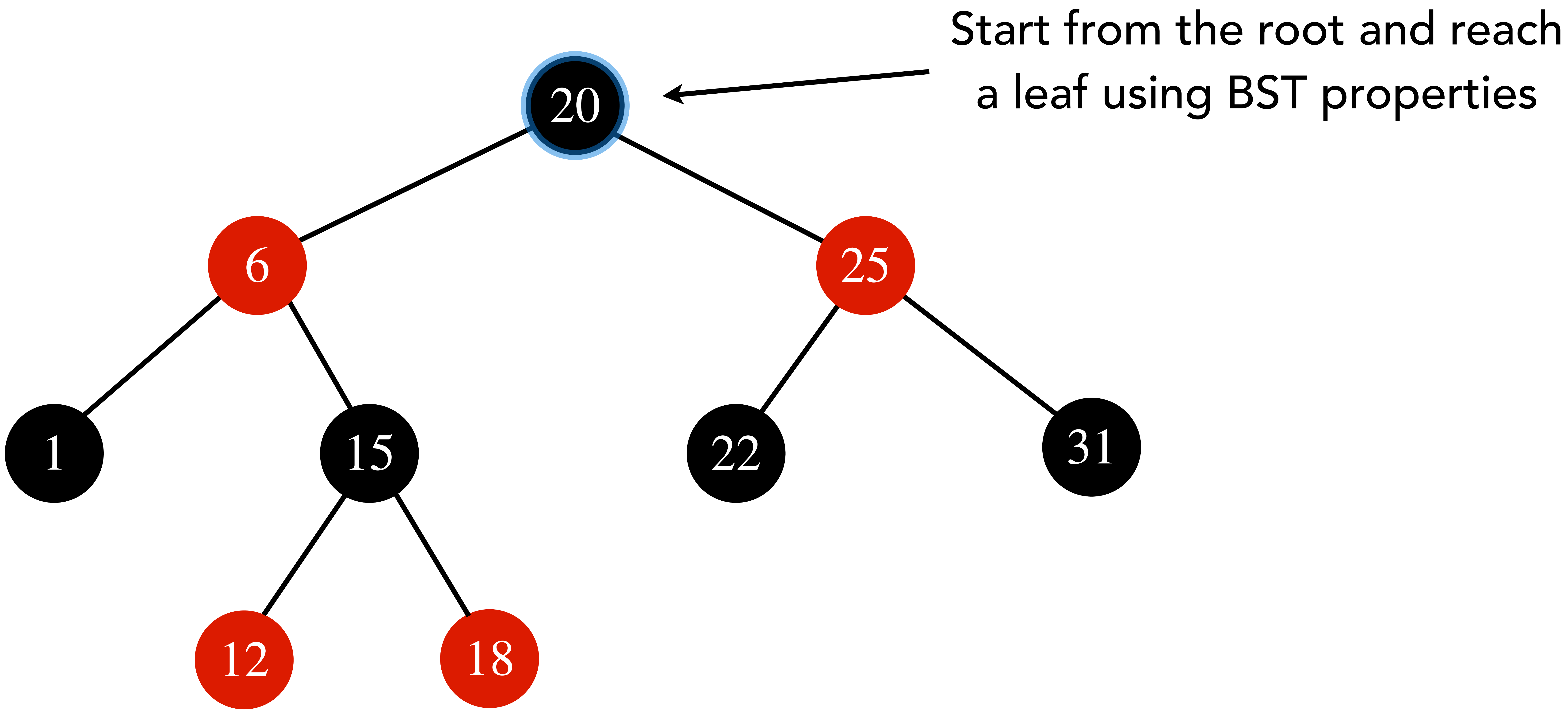
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



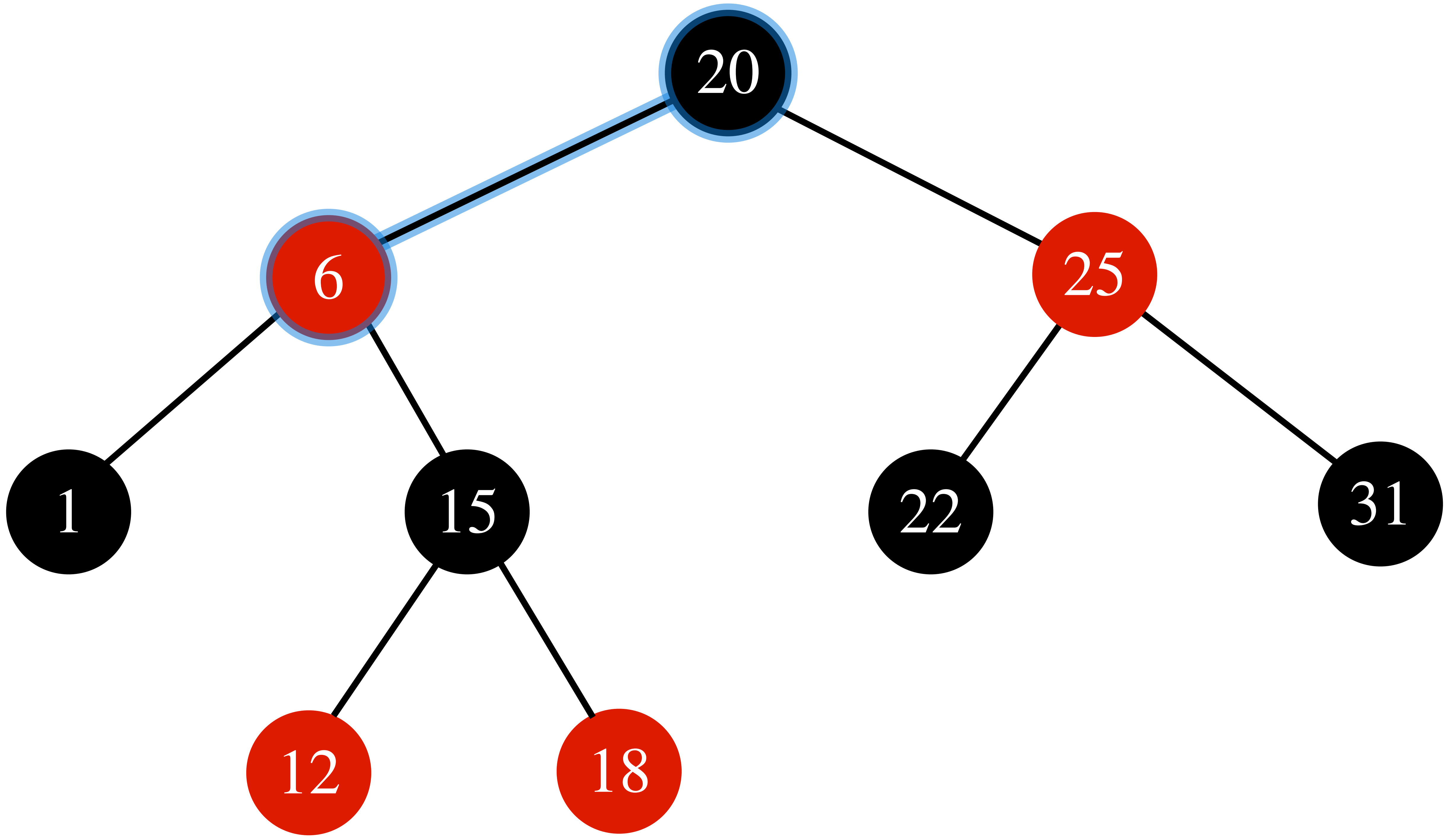
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



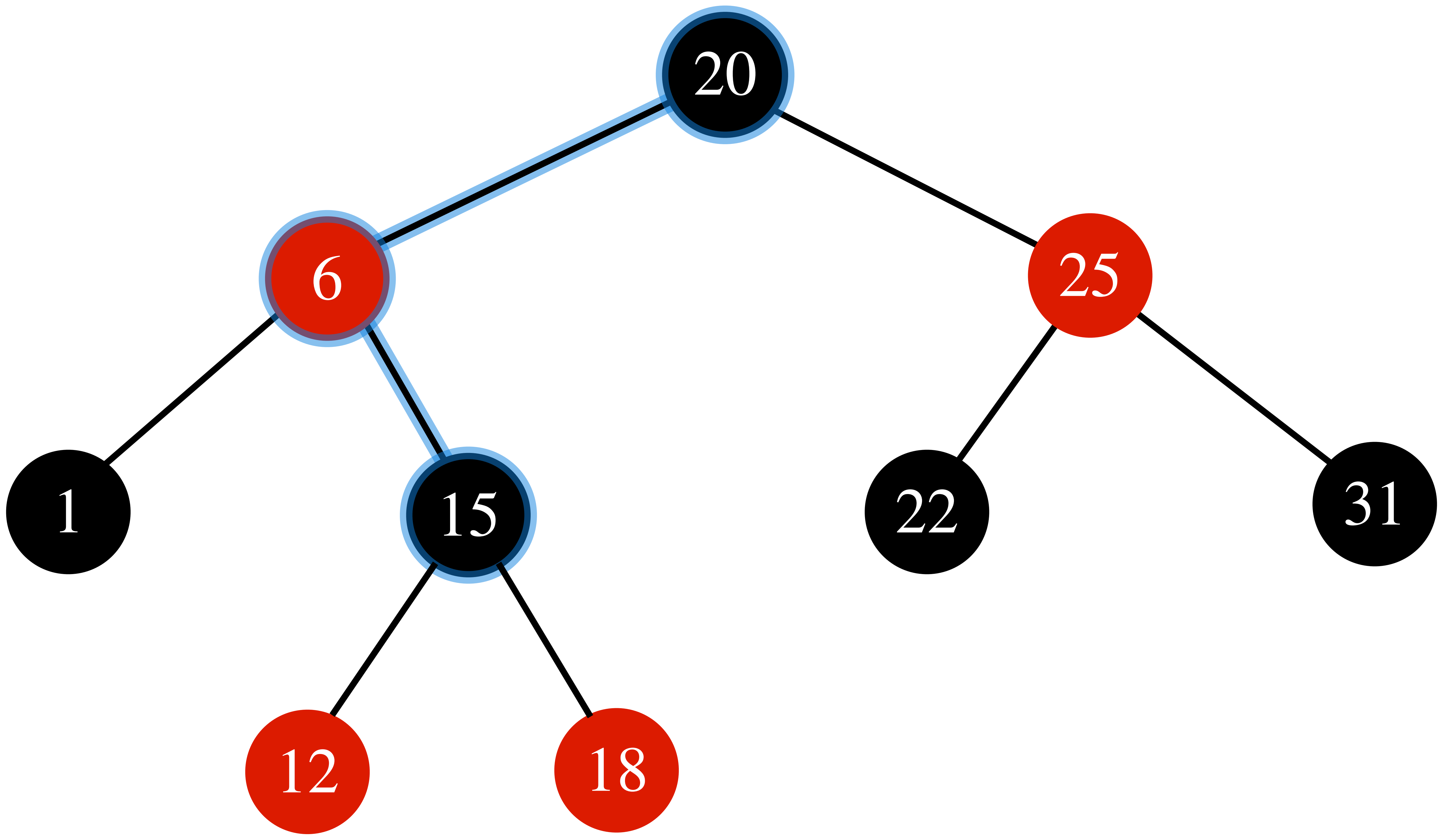
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



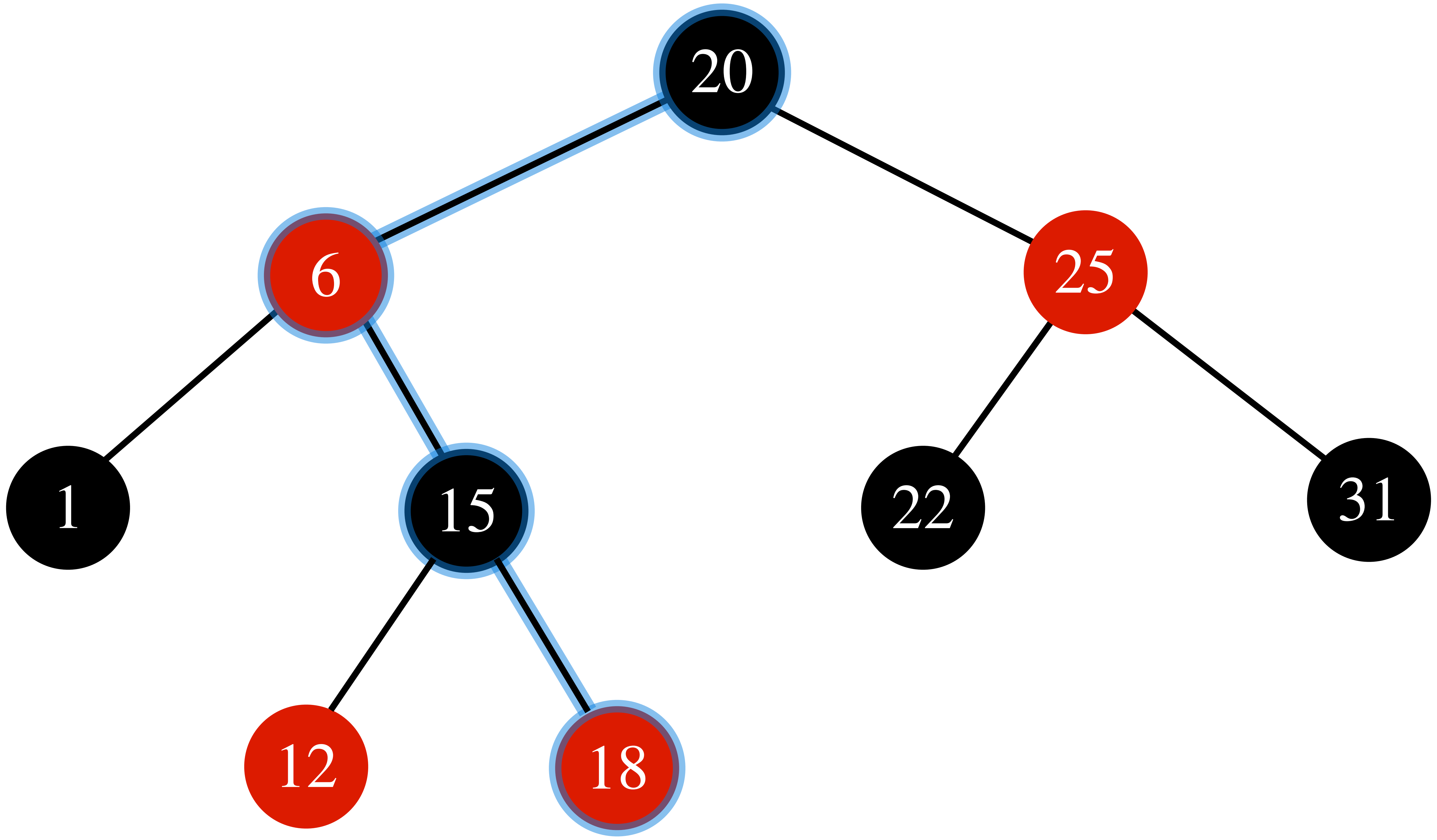
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



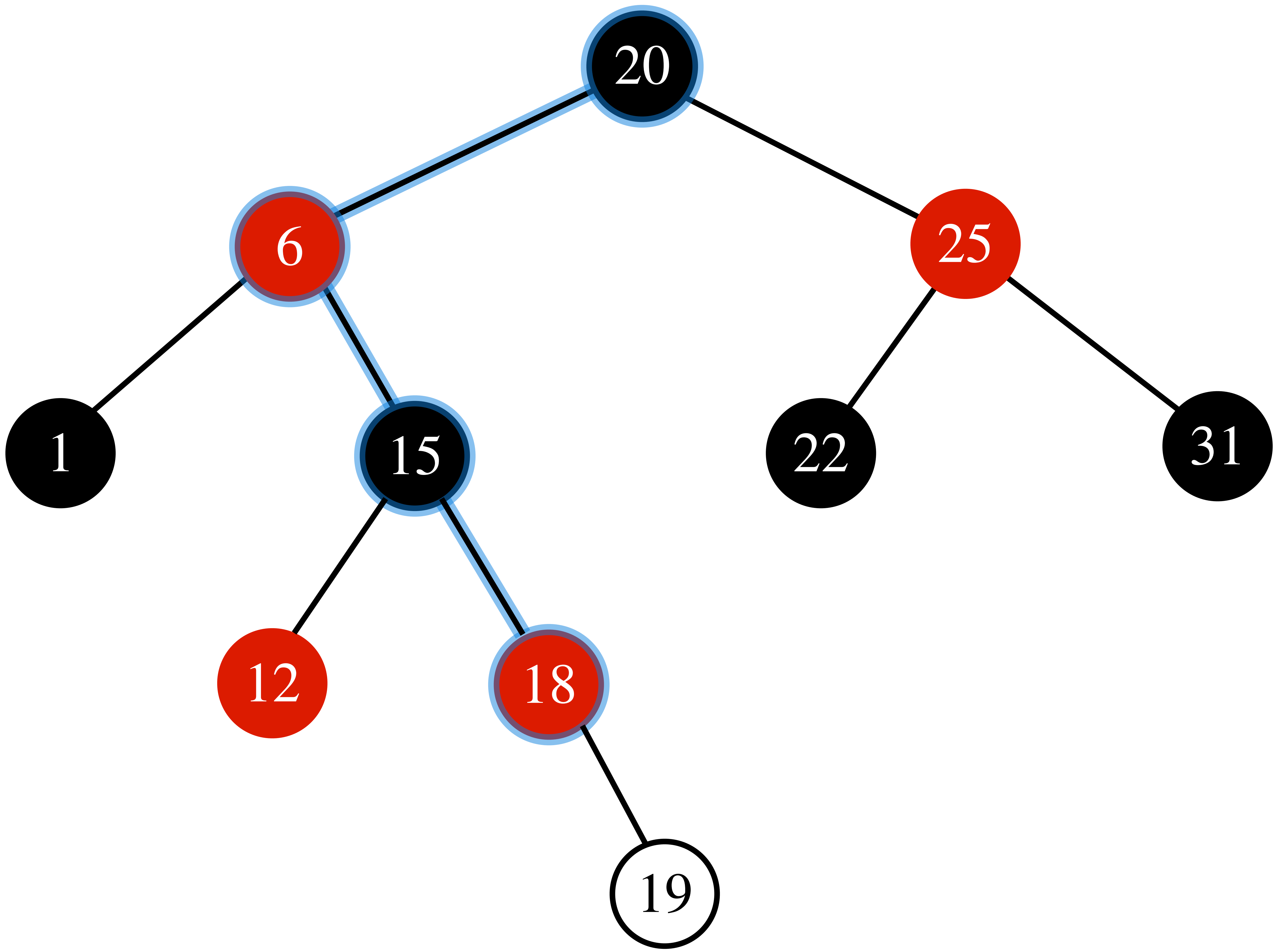
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



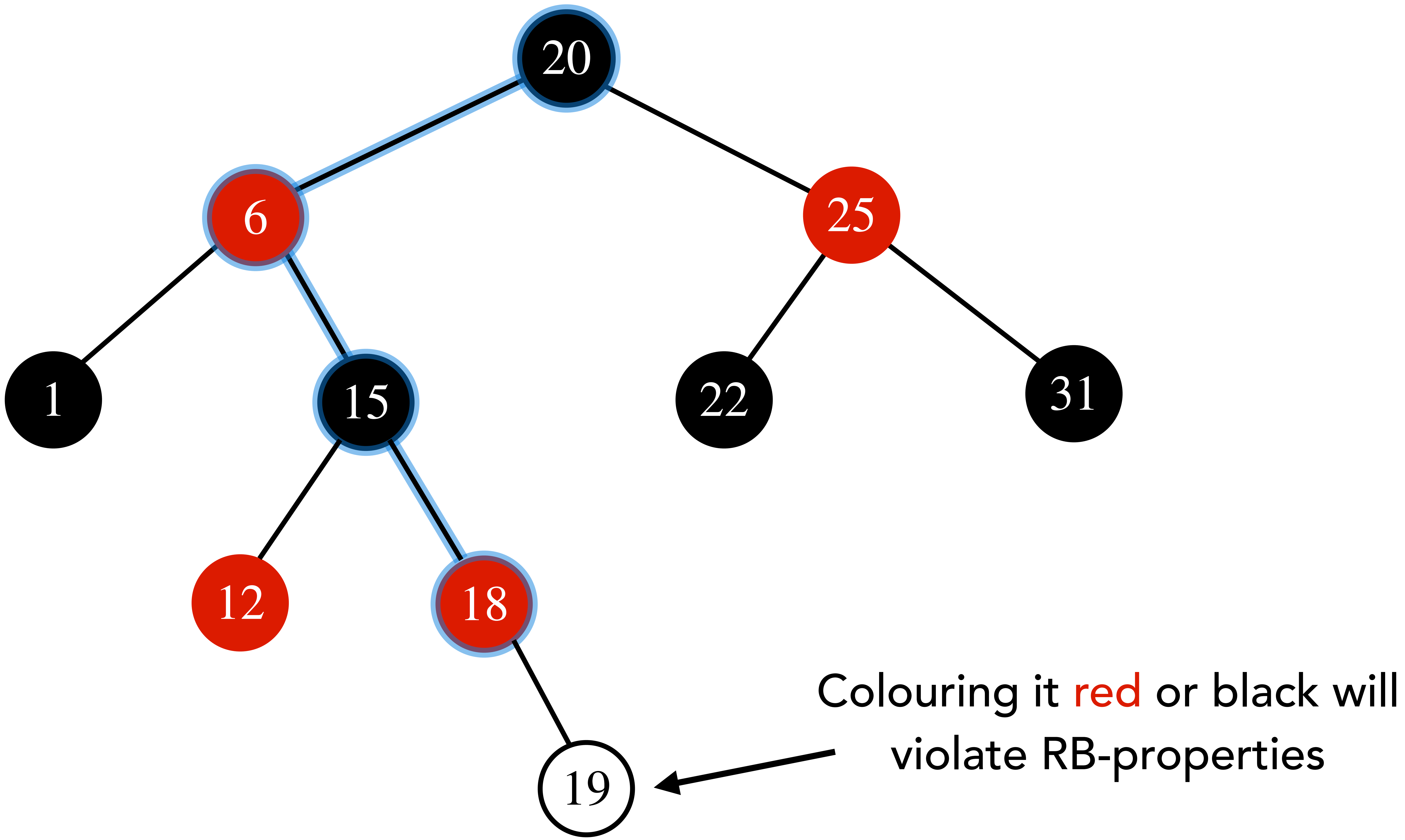
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



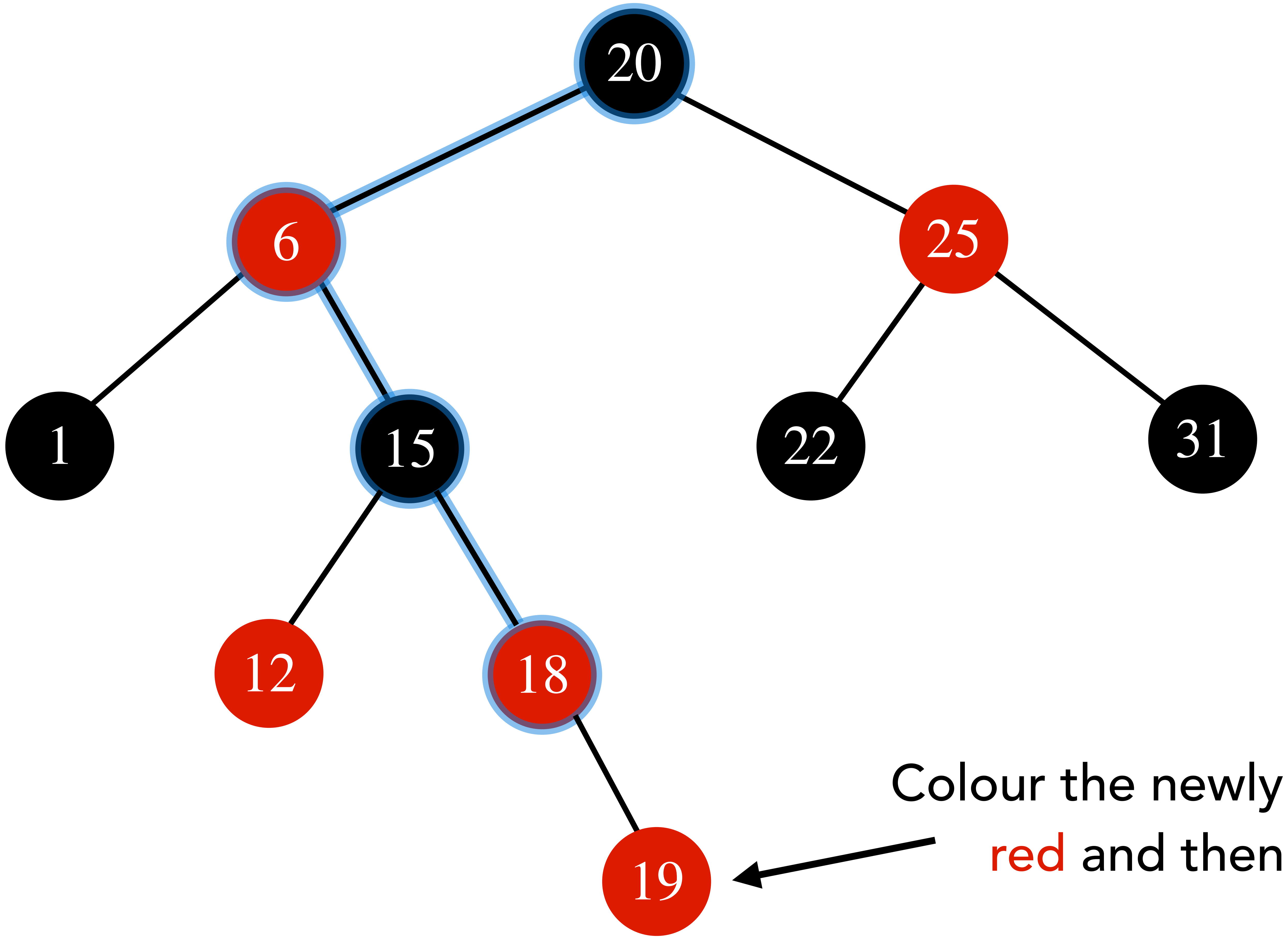
RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



RB-Trees: Insertion

Suppose we want to insert 19 in the following RB-tree:



Colour the newly inserted node red and then do fix-up.

RB-Trees: Insertion

RB-Trees: Insertion

Two stages of insertion:

RB-Trees: Insertion

Two stages of insertion:

- Insert the new node as it is done in a BST and colour the new node red.

RB-Trees: Insertion

Two stages of insertion:

- Insert the new node as it is done in a BST and colour the new node red.
- Do fix-ups as parent of the new node may also be a red node.