

Lecture 7

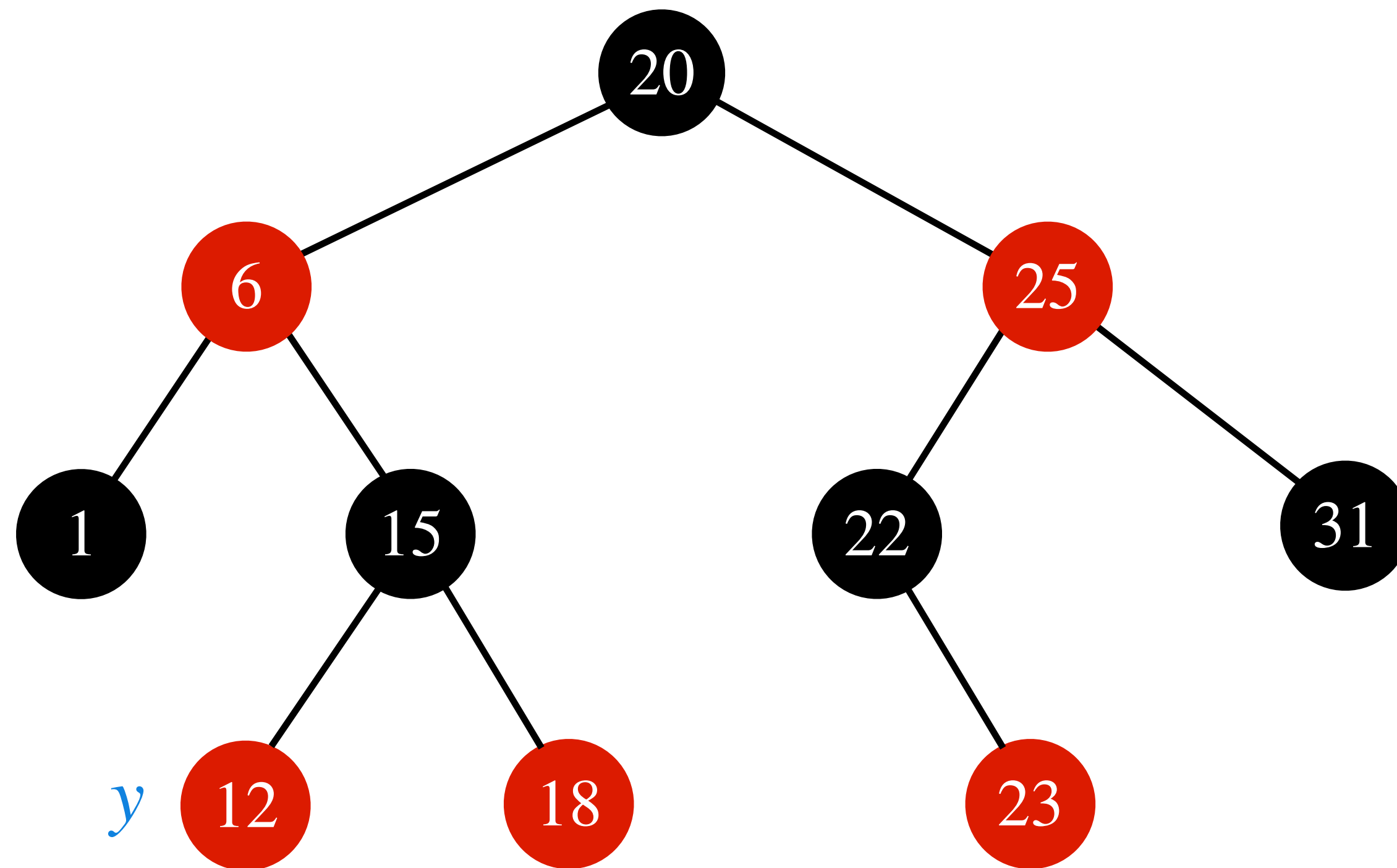
Red-Black Trees: Deletion

No Fix Ups When y is Red

We will see now that removing y when y is red causes no violation of RB-tree properties.

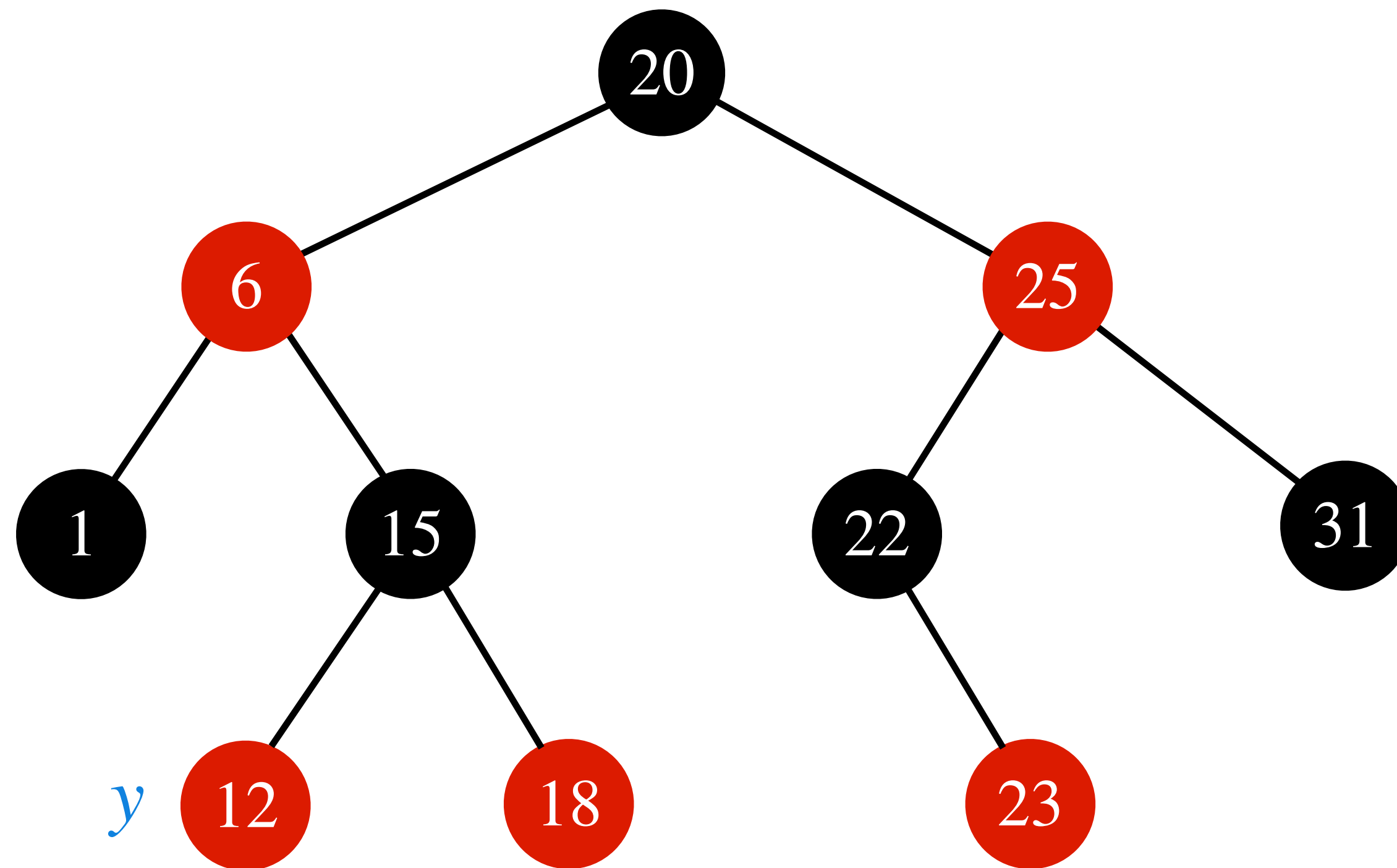
No Fix Ups When y is Red

Property 1: Every node has a colour either **red** or black.



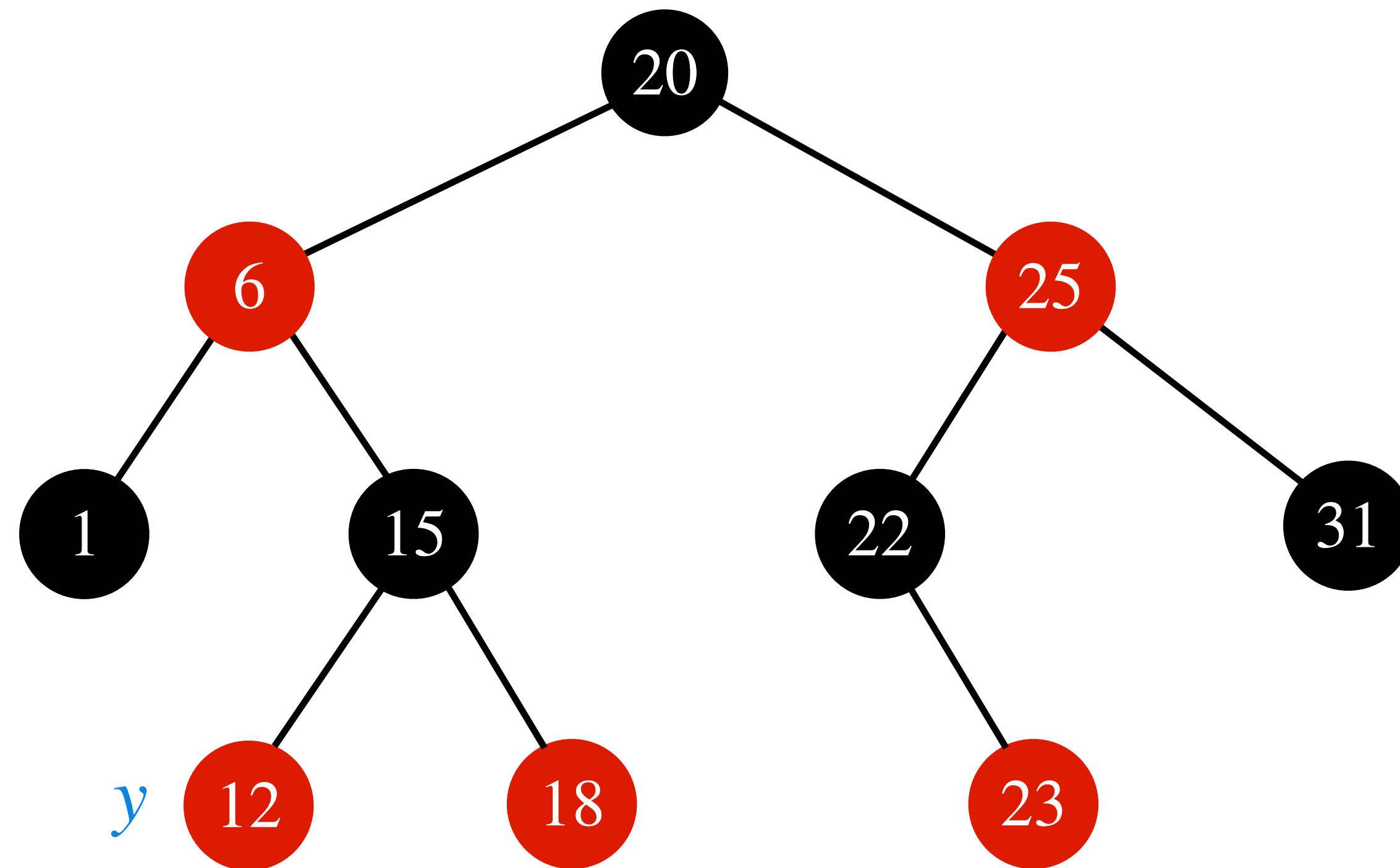
No Fix Ups When y is Red

Property 2: Root is black.



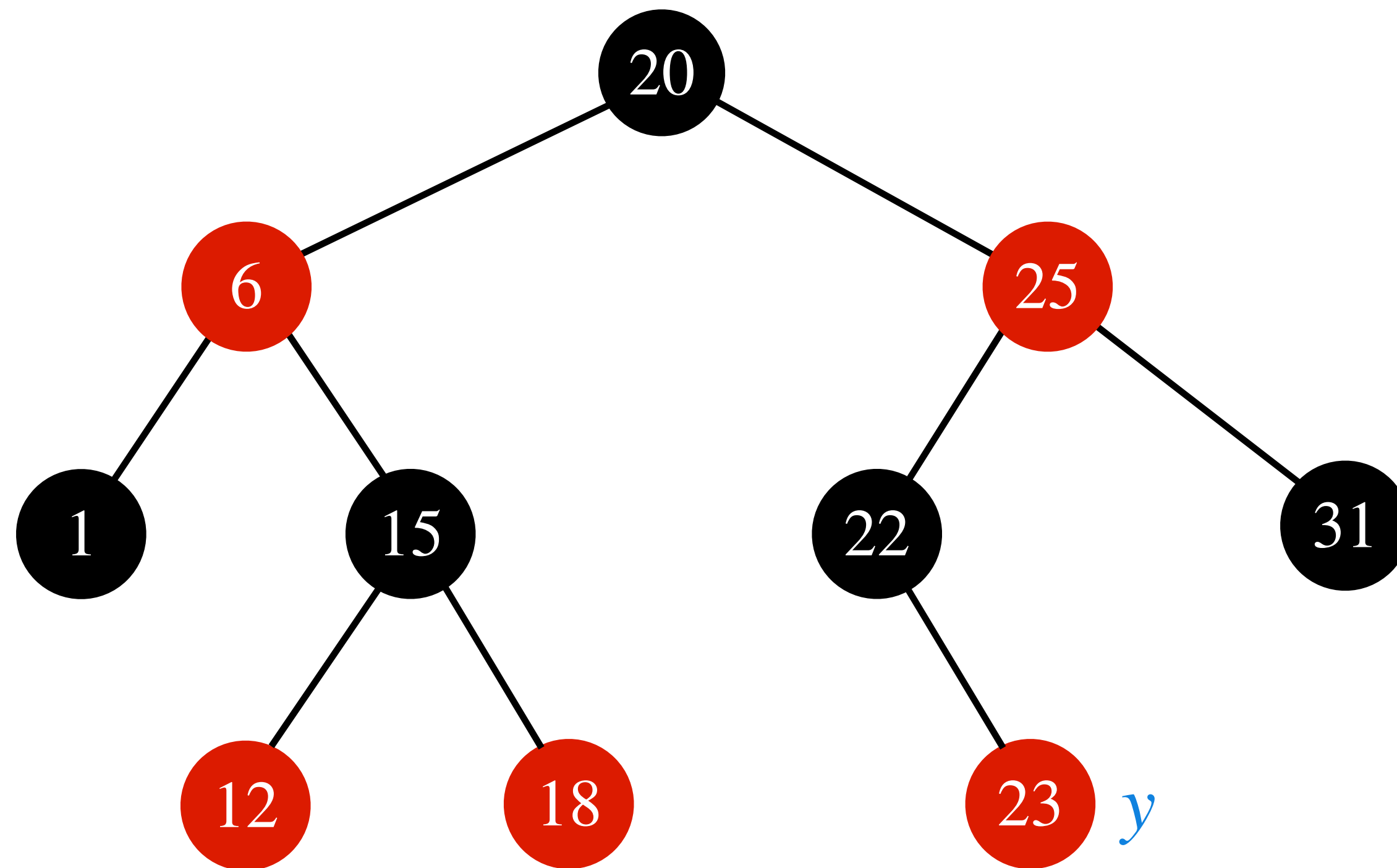
No Fix Ups When y is Red

Property 3: Leaf nodes are NIL nodes which are black in colour.



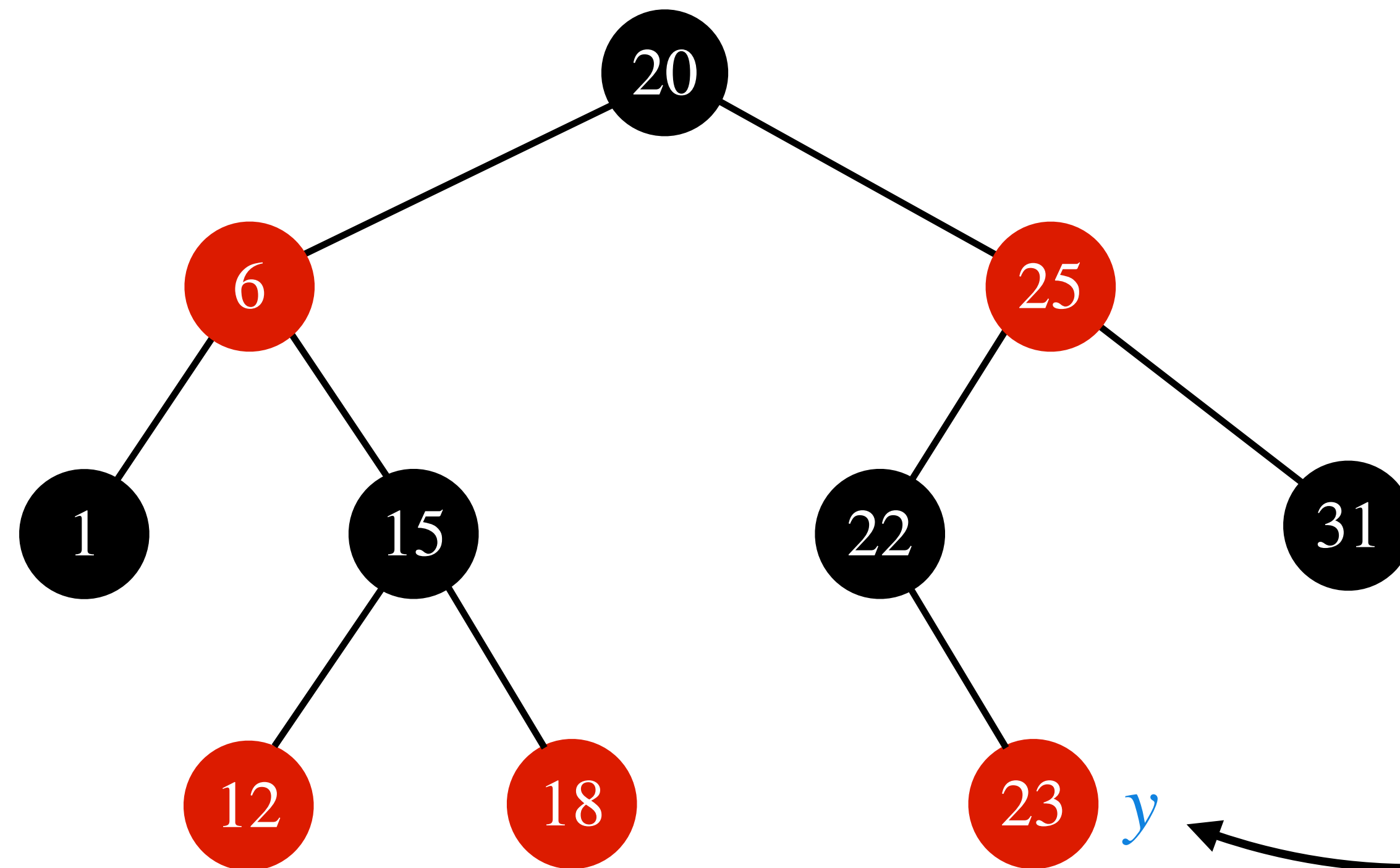
No Fix Ups When y is Red

Property 4: If a node is **red**, then both its children are black.



No Fix Ups When y is Red

Property 4: If a node is **red**, then both its children are black.

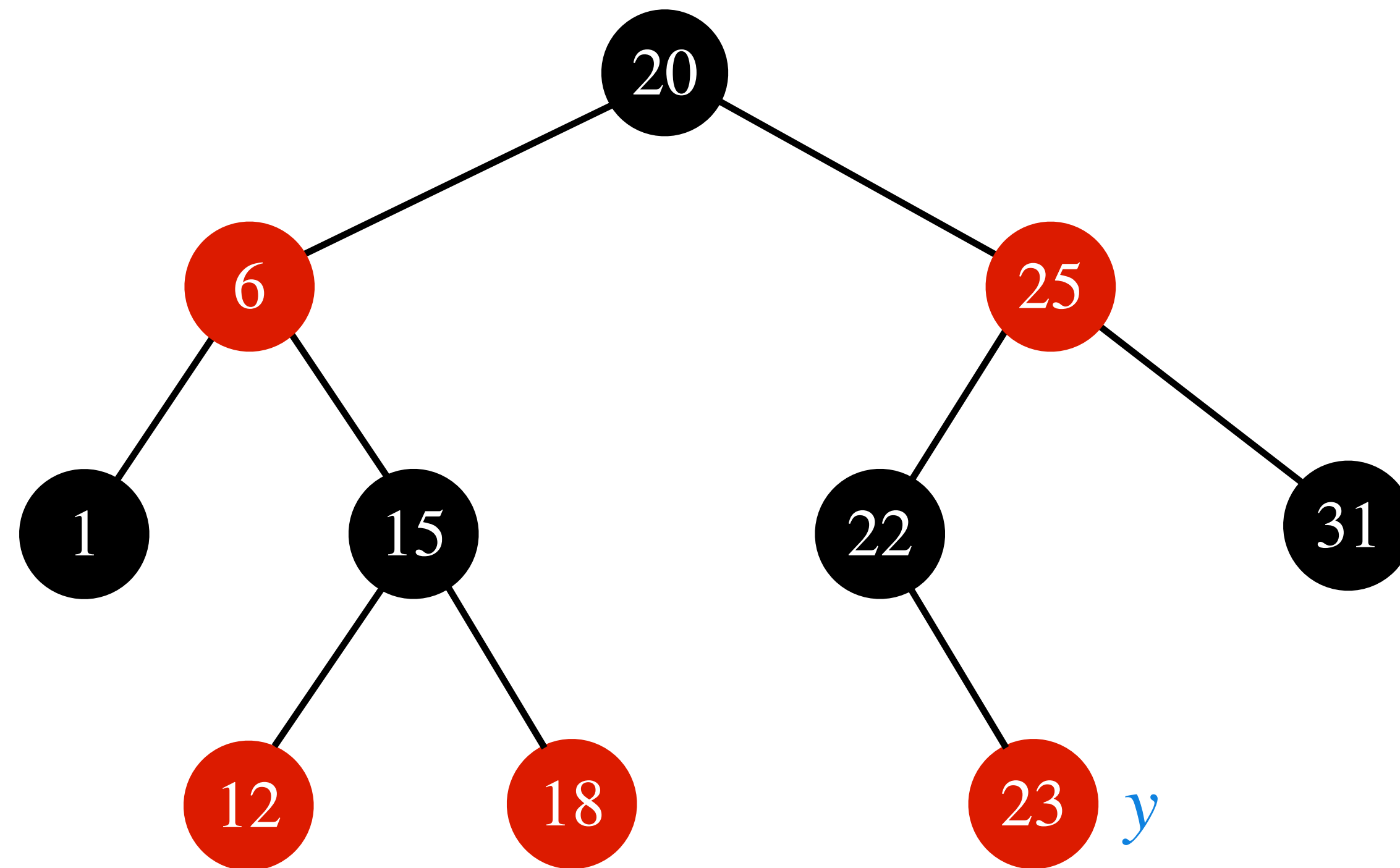


y 's parent and child
have to be black.

Hence, removing y doesn't
affect property 4.

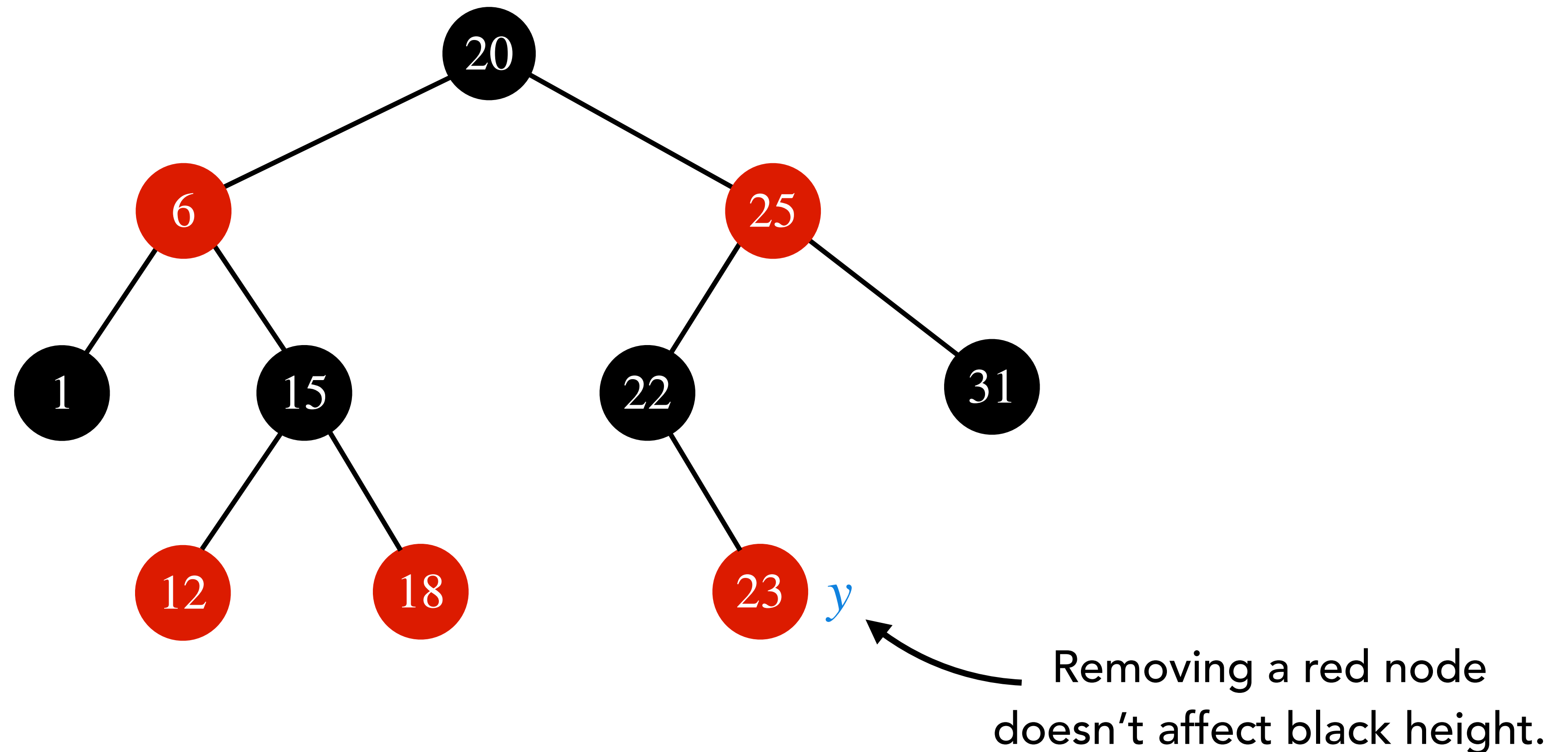
No Fix Ups When y is Red

Property 5: Black-height consistency...



No Fix Ups When y is Red

Property 5: Black-height consistency...

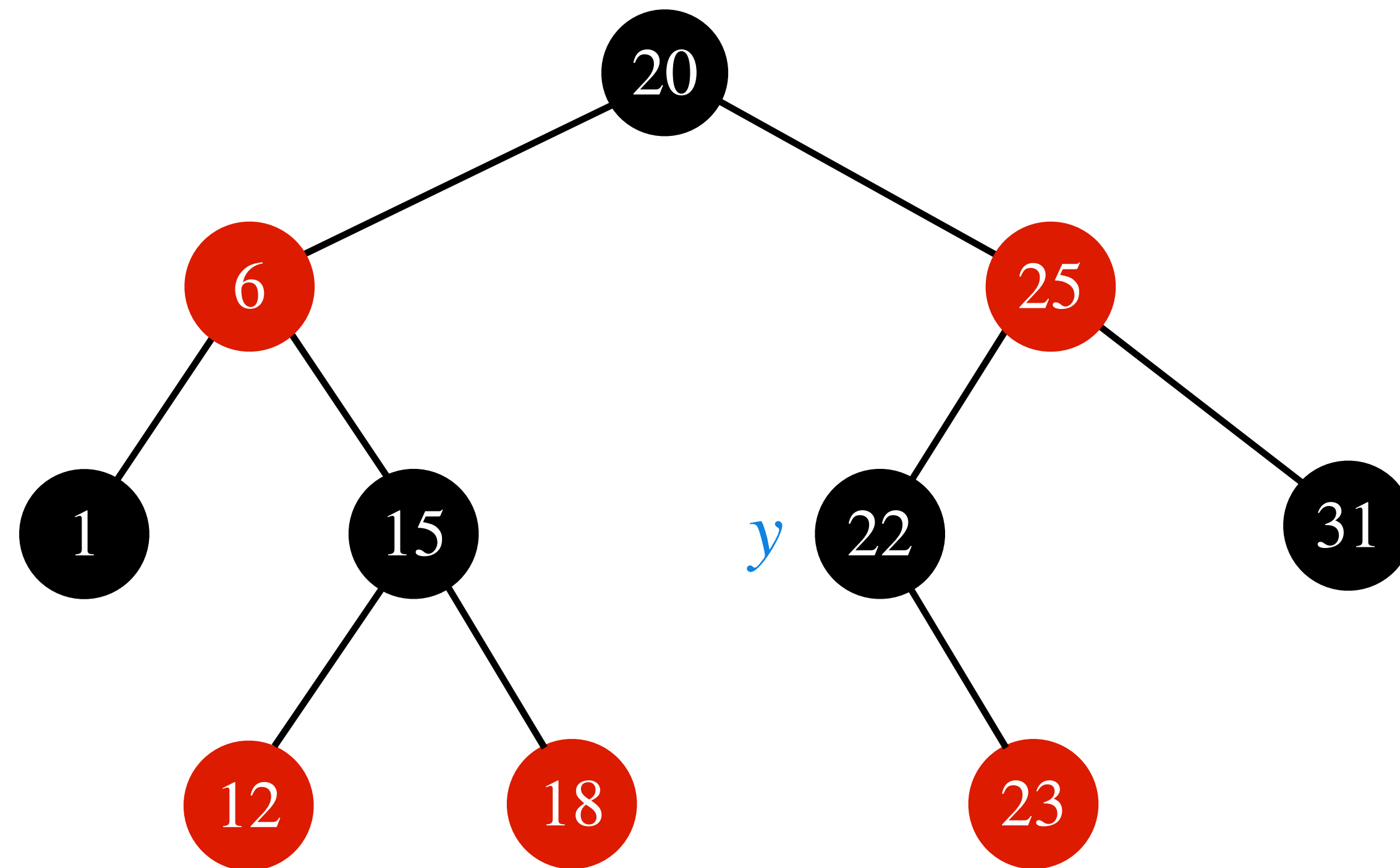


Fix Ups Required When y is Black

Let's see now what RB-tree properties will be violated on removing y when y is black.

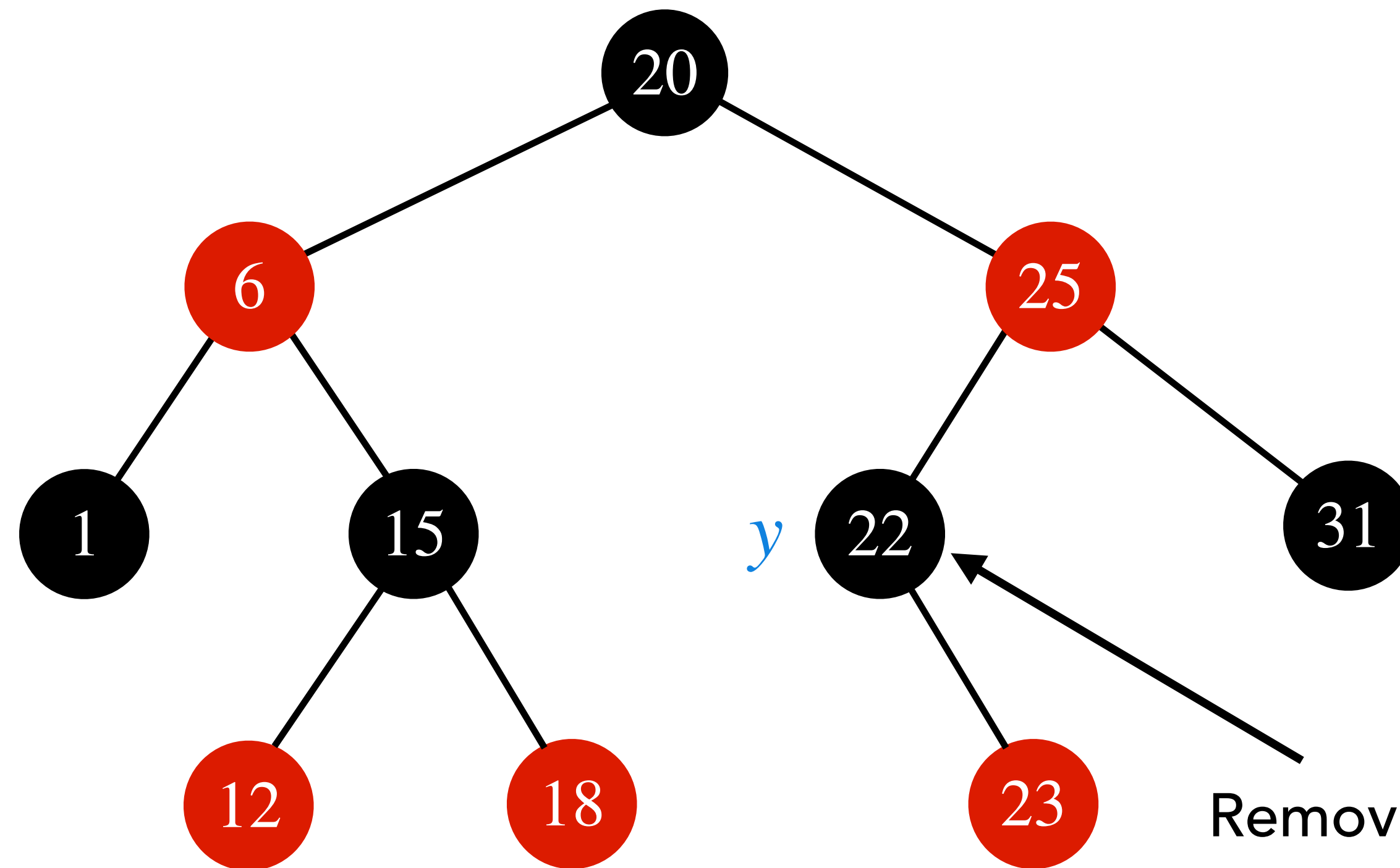
Fix Ups Required When y is Black

Property 1: Every node has a colour either **red** or black.



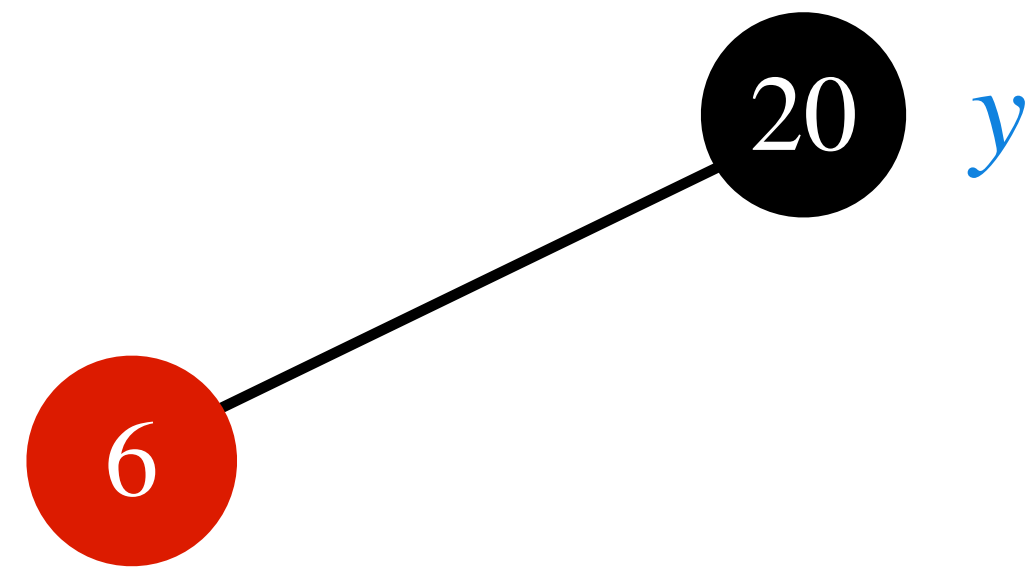
Fix Ups Required When y is Black

Property 1: Every node has a colour either **red** or black.



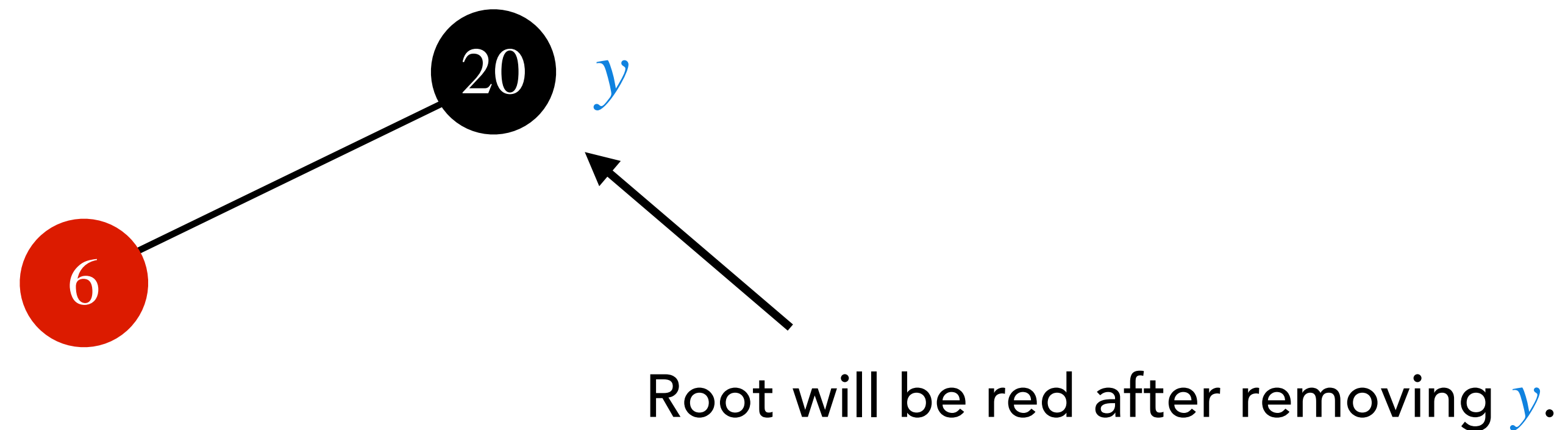
Fix Ups Required When y is Black

Property 2: Root is black.



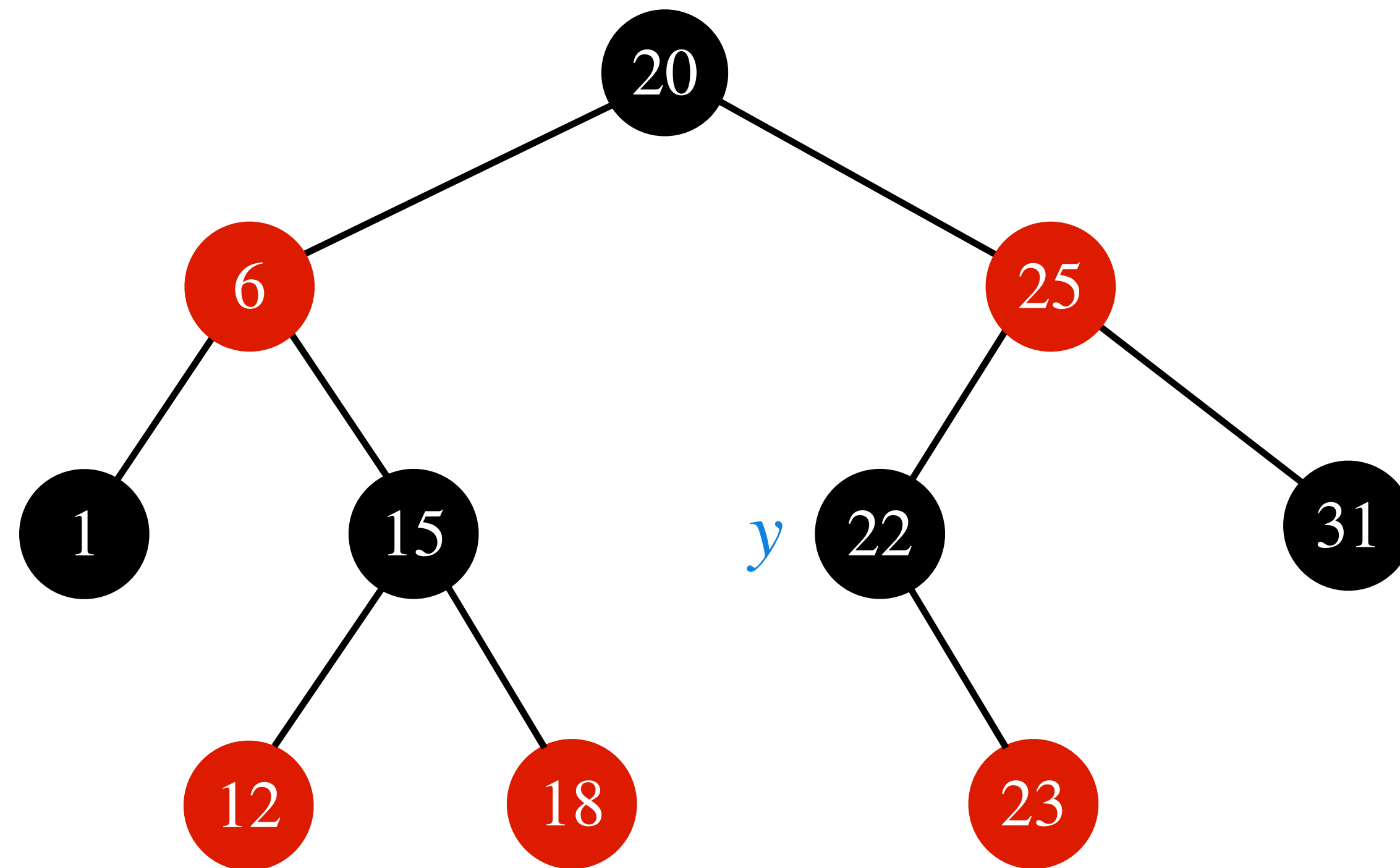
Fix Ups Required When y is Black

Property 2: Root is black.



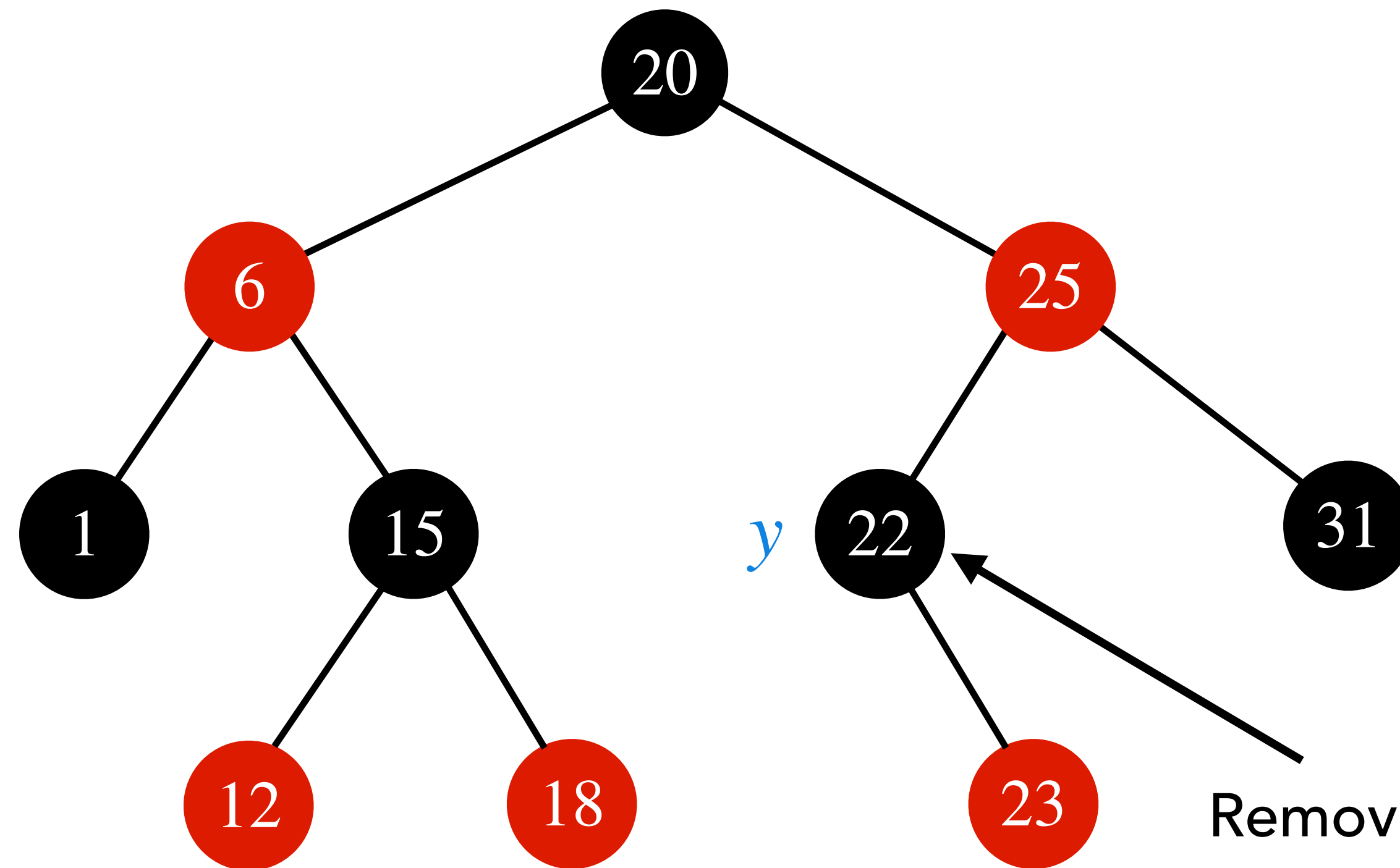
Fix Ups Required When y is Black

Property 3: Leaf nodes are NIL nodes which are black in colour.



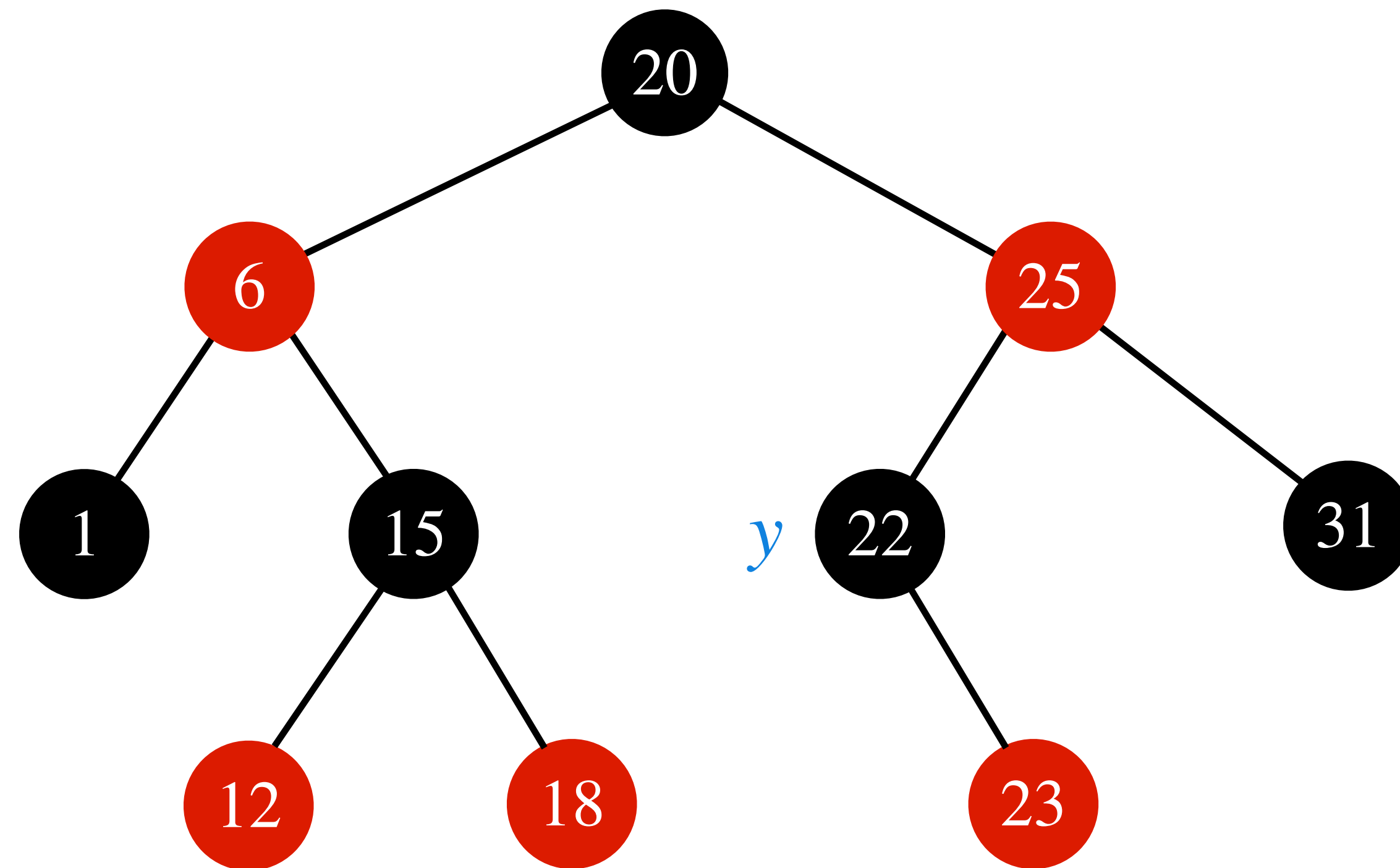
Fix Ups Required When y is Black

Property 3: Leaf nodes are NIL nodes which are black in colour.



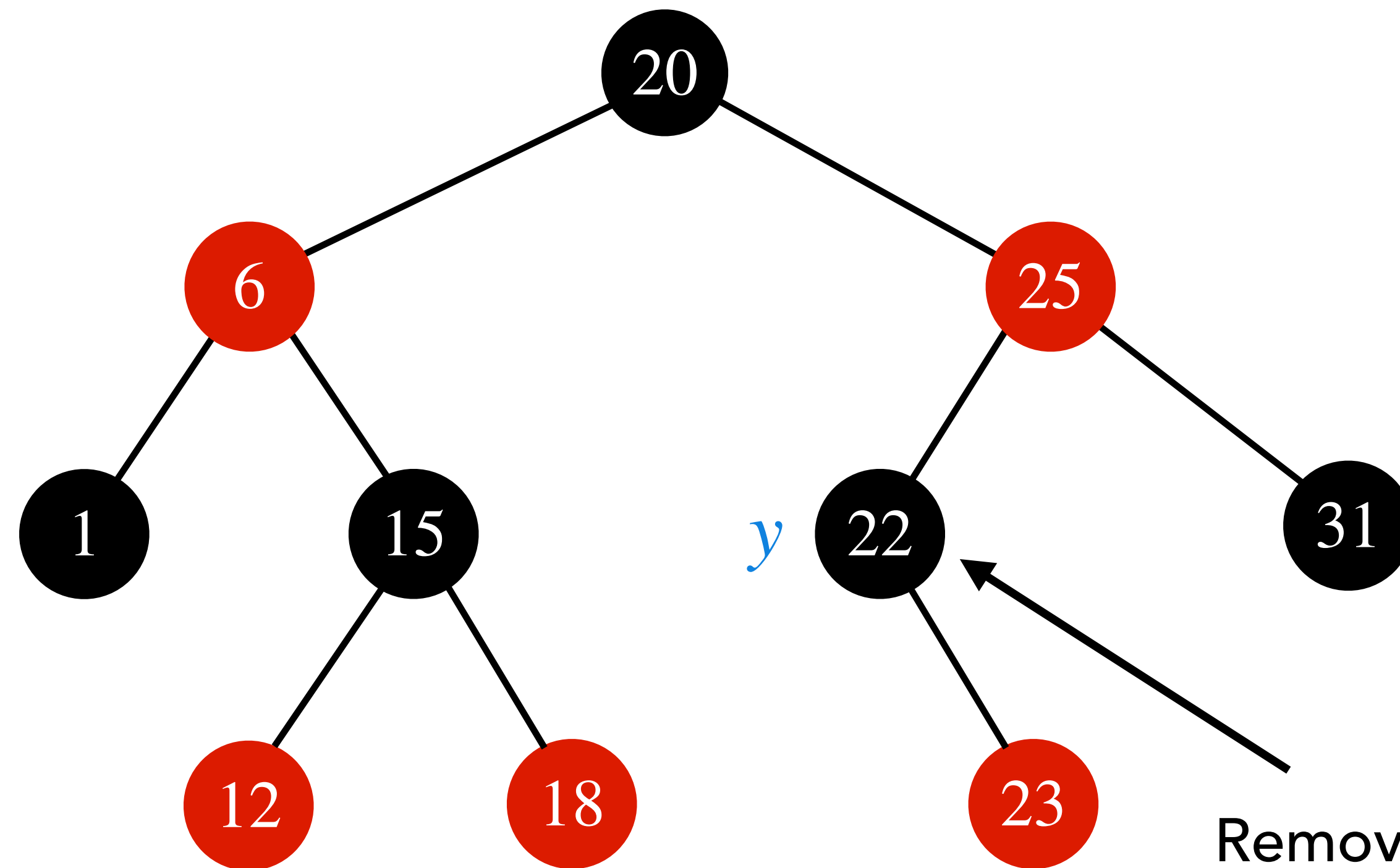
Fix Ups Required When y is Black

Property 4: If a node is **red**, then both its children are black.



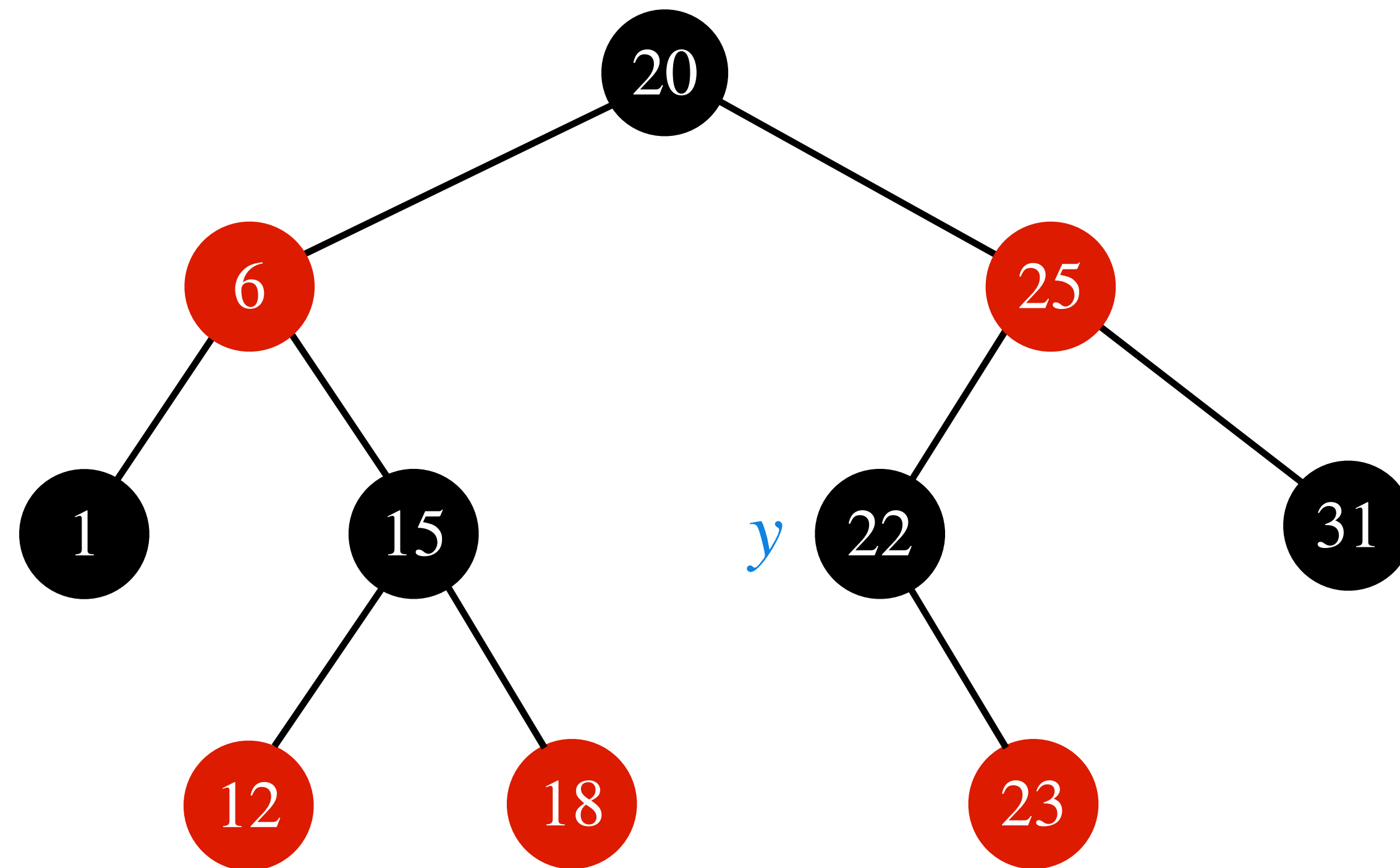
Fix Ups Required When y is Black

Property 4: If a node is **red**, then both its children are black.



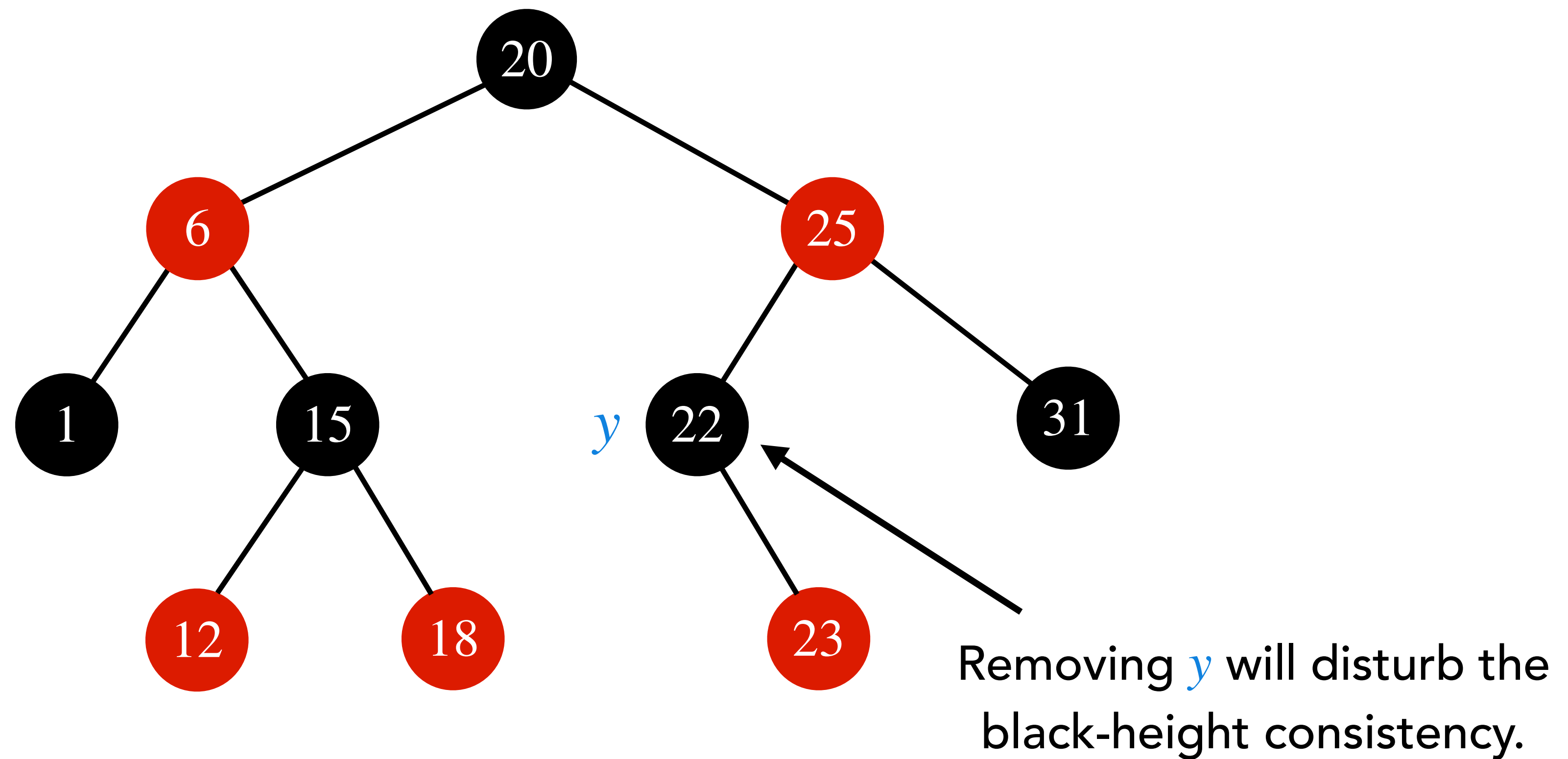
Fix Ups Required When y is Black

Property 5: Black-height consistency...



Fix Ups Required When y is Black

Property 5: Black-height consistency...



Fix Ups Required When y is Black

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

- **Property 2:** Root is black.

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

- **Property 2:** Root is black.
- **Property 4:** If a node is **red**, then both its children are black.

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

- **Property 2:** Root is black.
- **Property 4:** If a node is **red**, then both its children are black.
- **Property 5:** Black-height consistency...

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

- **Property 2:** Root is black.
- **Property 4:** If a node is **red**, then both its children are black.
- **Property 5:** Black-height consistency...

Recall that:

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

- **Property 2:** Root is black.
- **Property 4:** If a node is **red**, then both its children are black.
- **Property 5:** Black-height consistency...

Recall that:

- y has at most one non-NIL child.

Fix Ups Required When y is Black

If y is black, then **deletion** can violate the following properties:

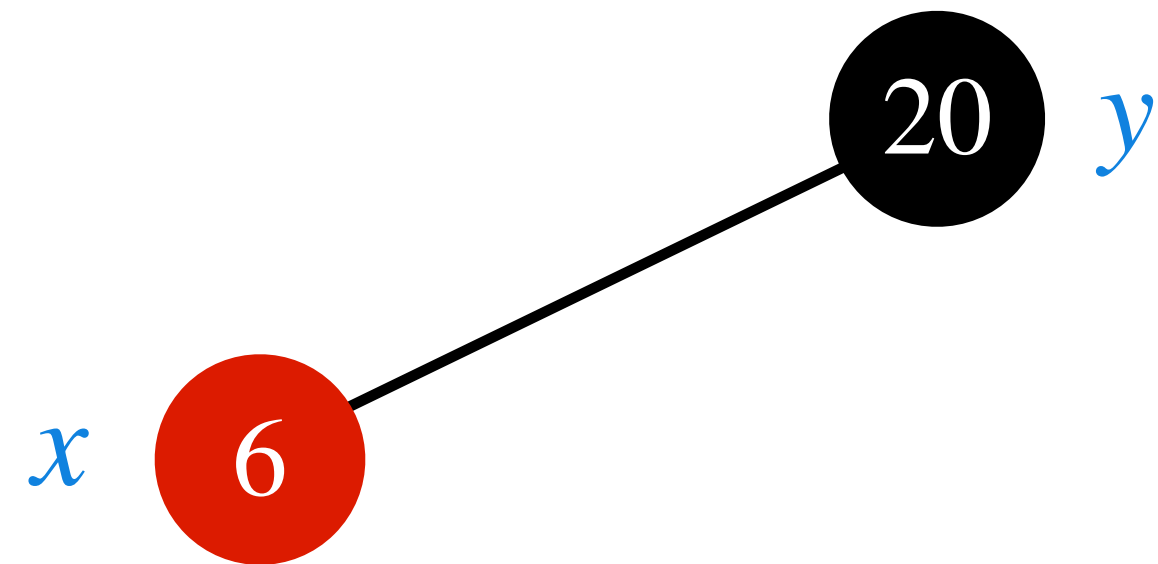
- **Property 2:** Root is black.
- **Property 4:** If a node is **red**, then both its children are black.
- **Property 5:** Black-height consistency...

Recall that:

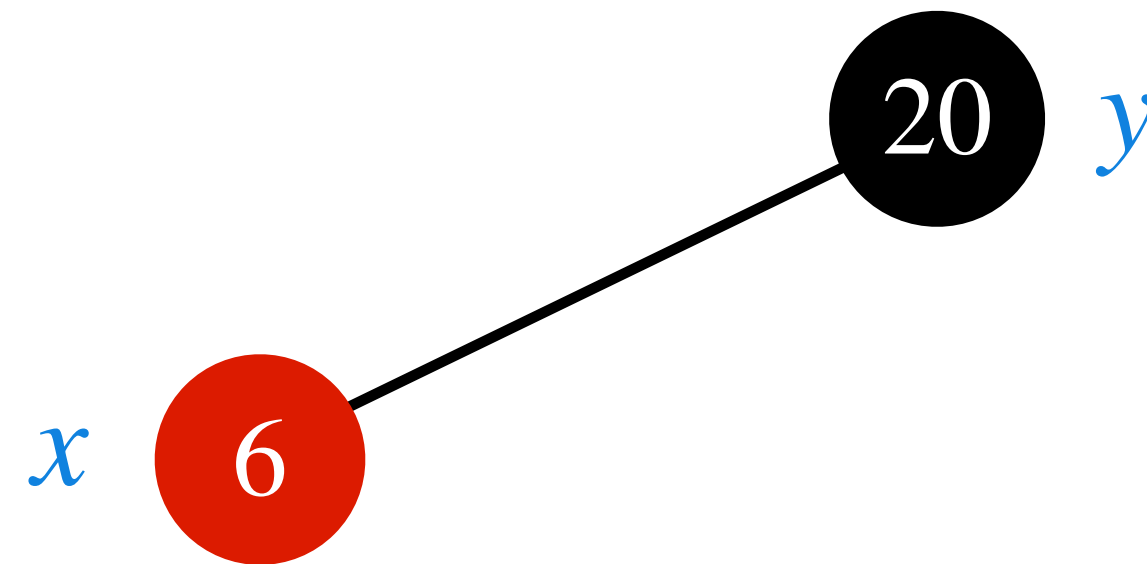
- y has at most one non-NIL child.
- x is y 's only non-NIL child or NIL node if y has no non-NIL children.

Fixing Violation of Property 2

Fixing Violation of Property 2

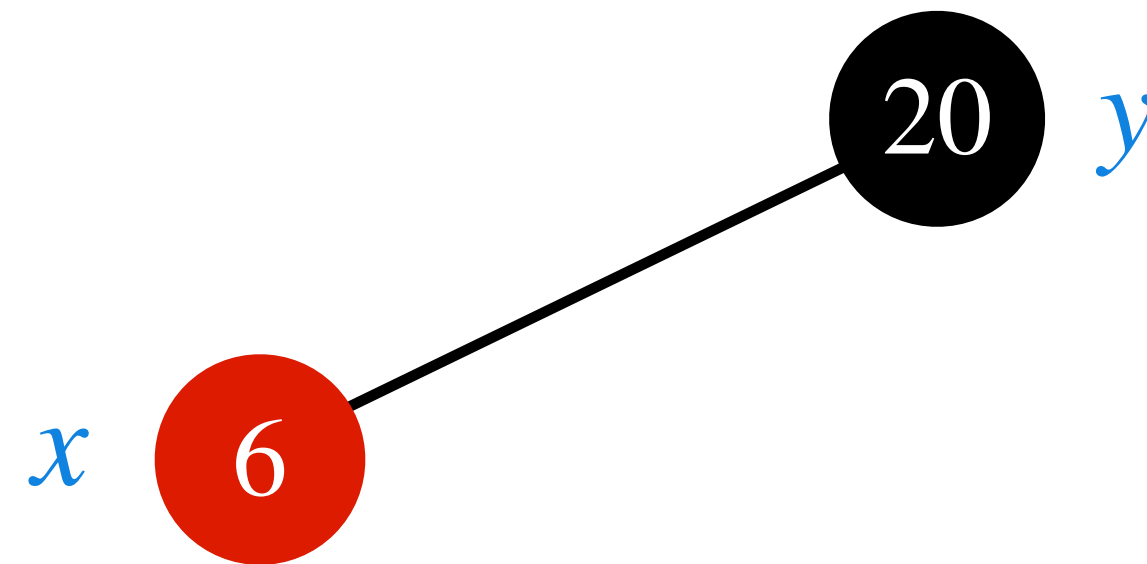


Fixing Violation of Property 2



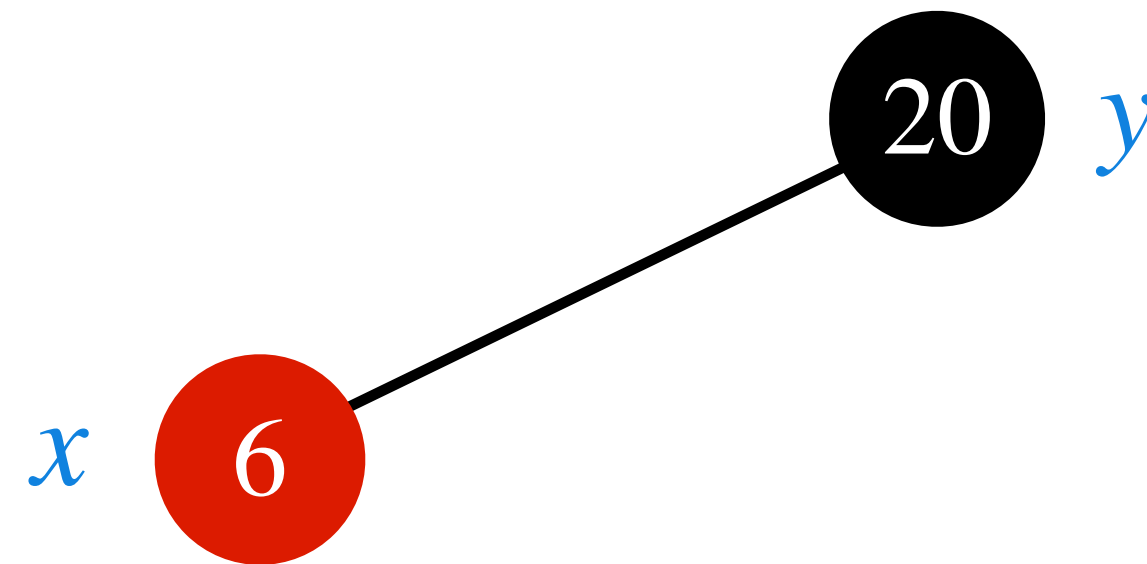
Can property 4 and 5 also be violated when property 2 is being violated?

Fixing Violation of Property 2



Can property 4 and 5 also be violated when property 2 is being violated? No.

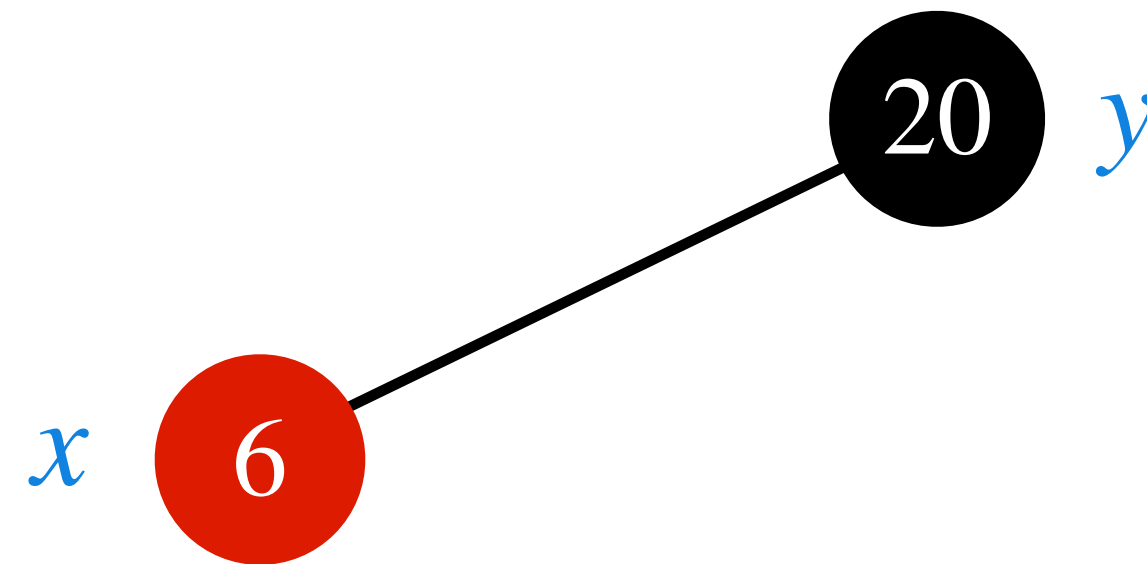
Fixing Violation of Property 2



Can property 4 and 5 also be violated when property 2 is being violated? No.

How can we fix the violation of property 2?

Fixing Violation of Property 2

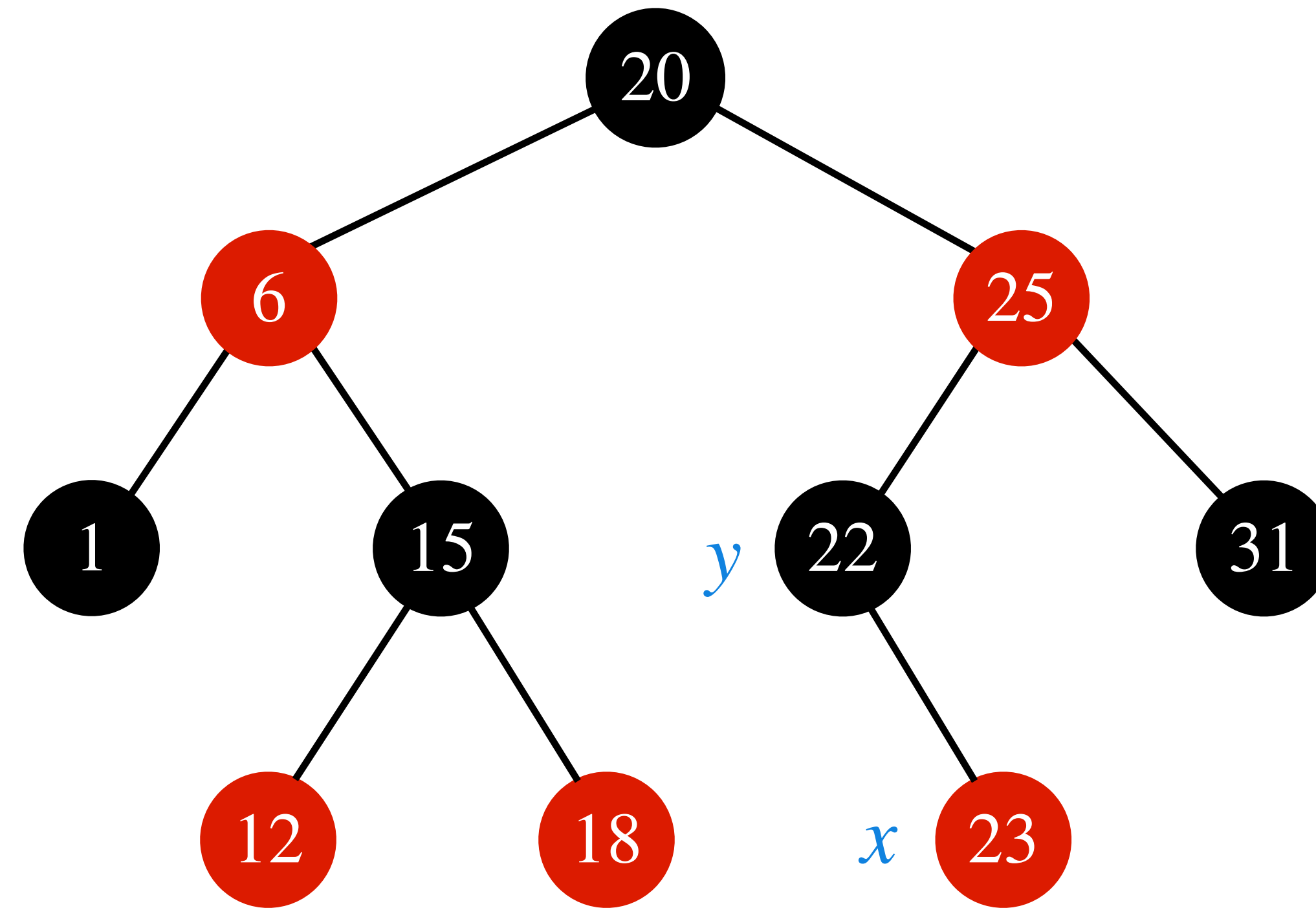


Can property 4 and 5 also be violated when property 2 is being violated? No.

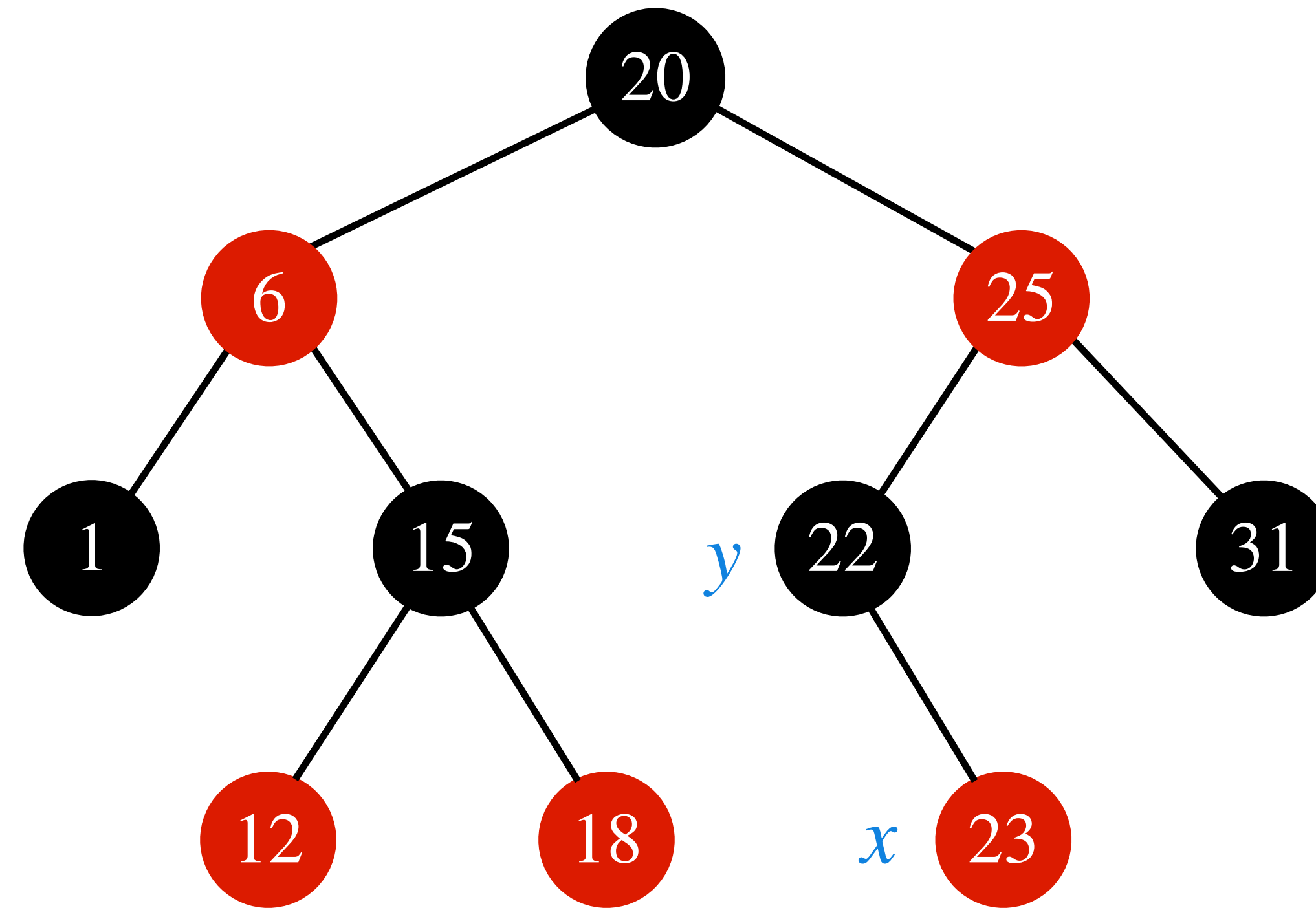
How can we fix the violation of property 2? Colour x black.

Fixing Violation of Property 4

Fixing Violation of Property 4

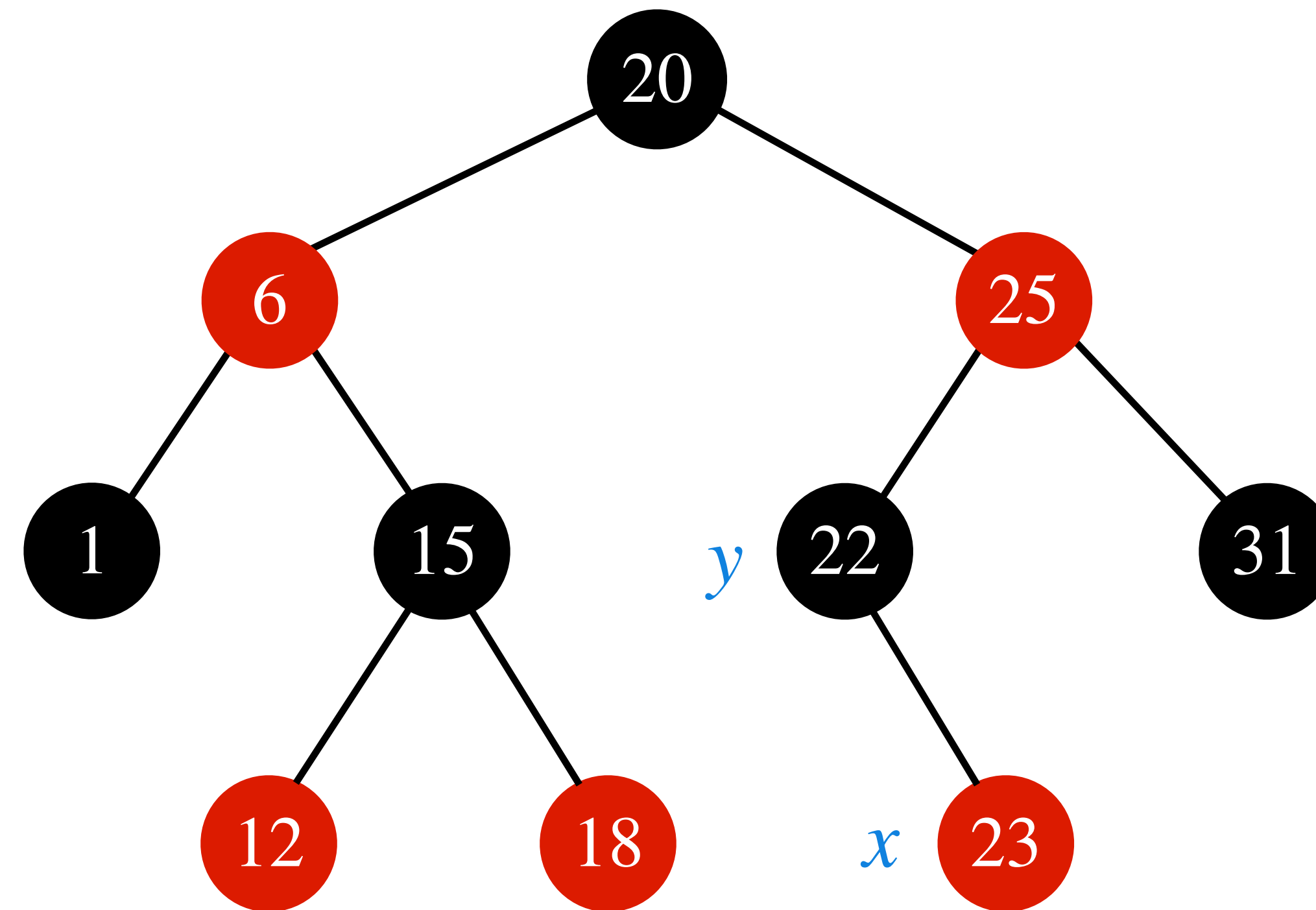


Fixing Violation of Property 4



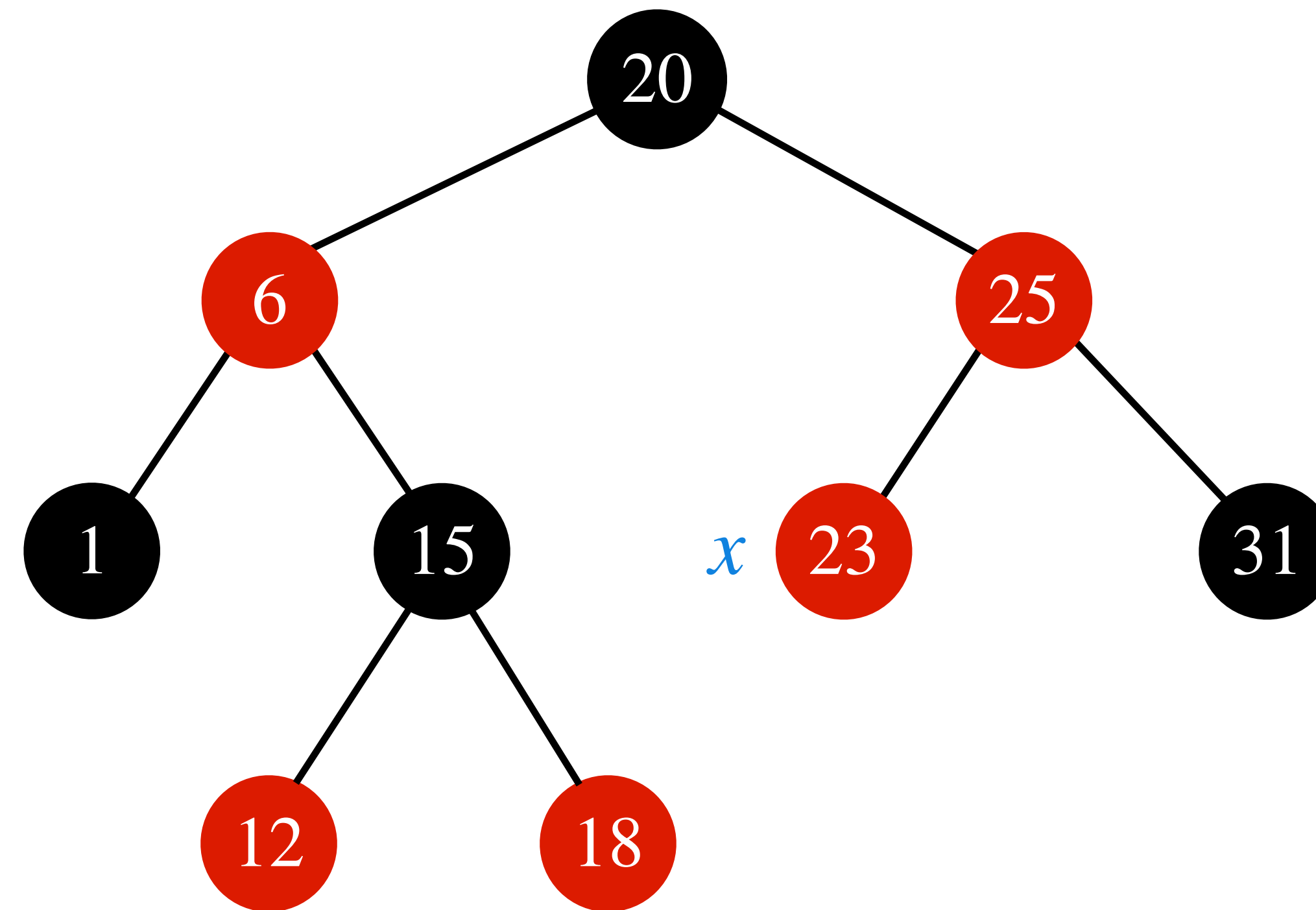
What other properties will also violate with property 4's violation?

Fixing Violation of Property 4



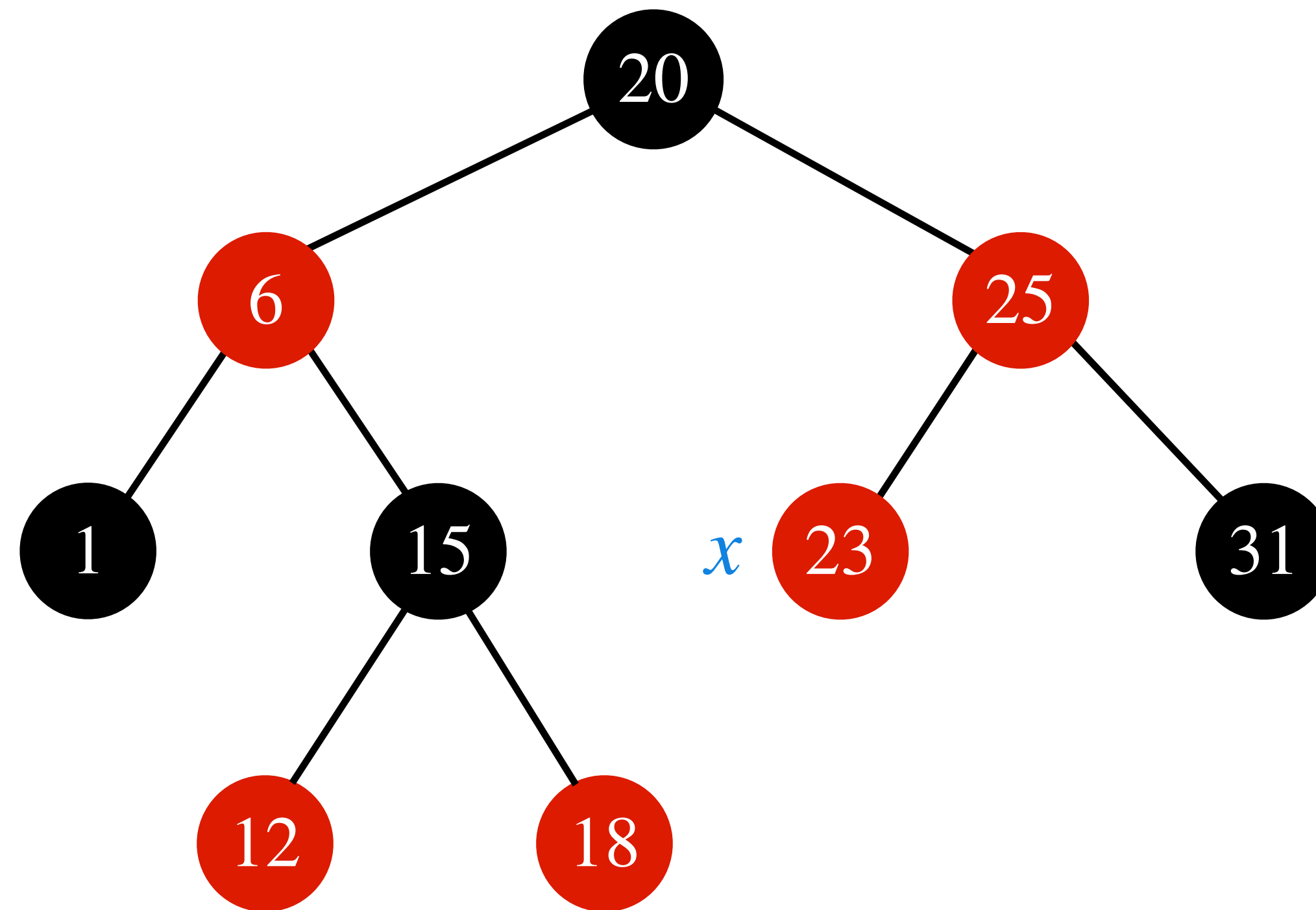
What other properties will also violate with property 4's violation? Only property 5.

Fixing Violation of Property 4



What other properties will also violate with property 4's violation? Only property 5.

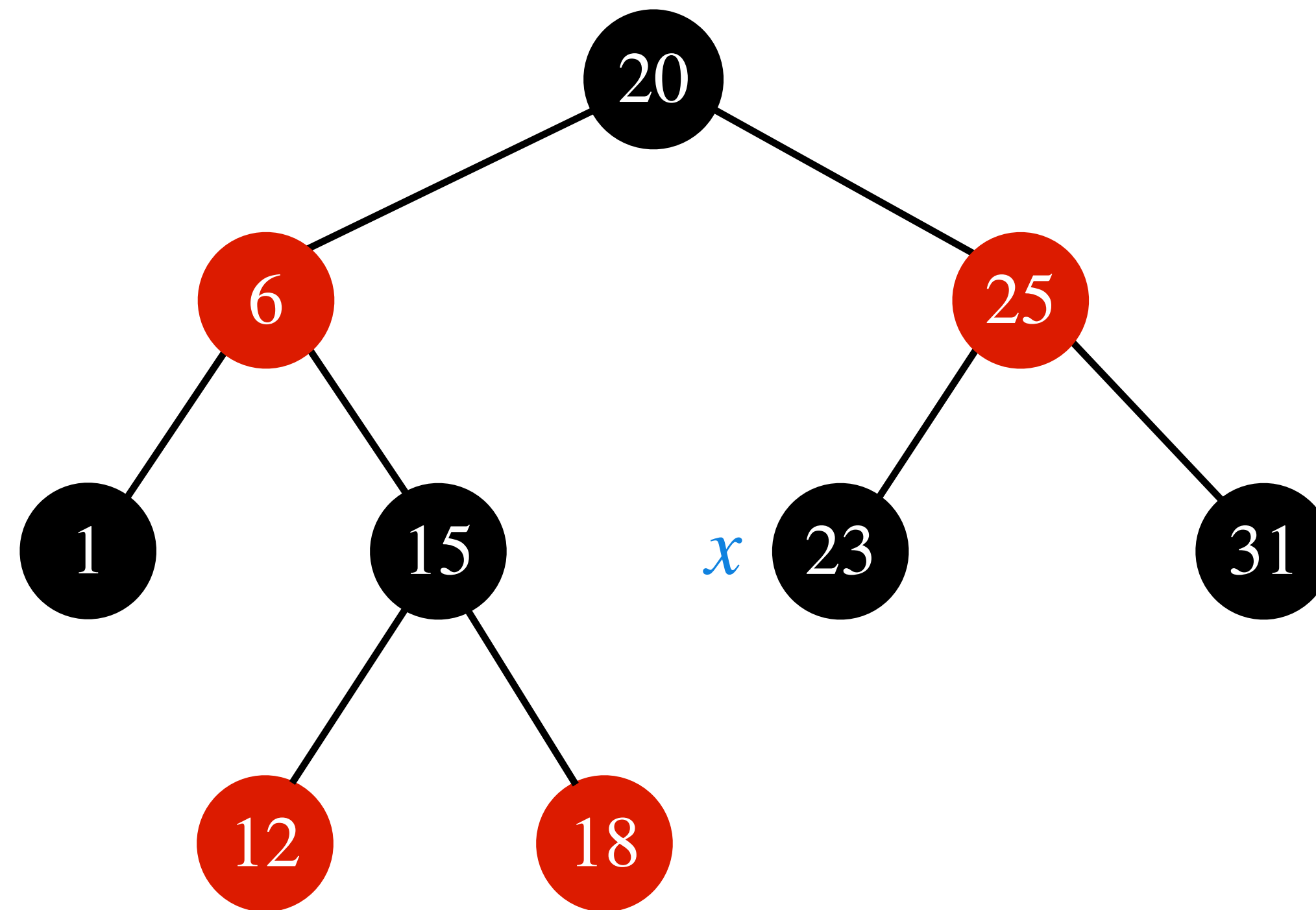
Fixing Violation of Property 4



What other properties will also violate with property 4's violation? Only property 5.

How can we fix the violation of property 4 and 5?

Fixing Violation of Property 4



What other properties will also violate with property 4's violation? Only property 5.

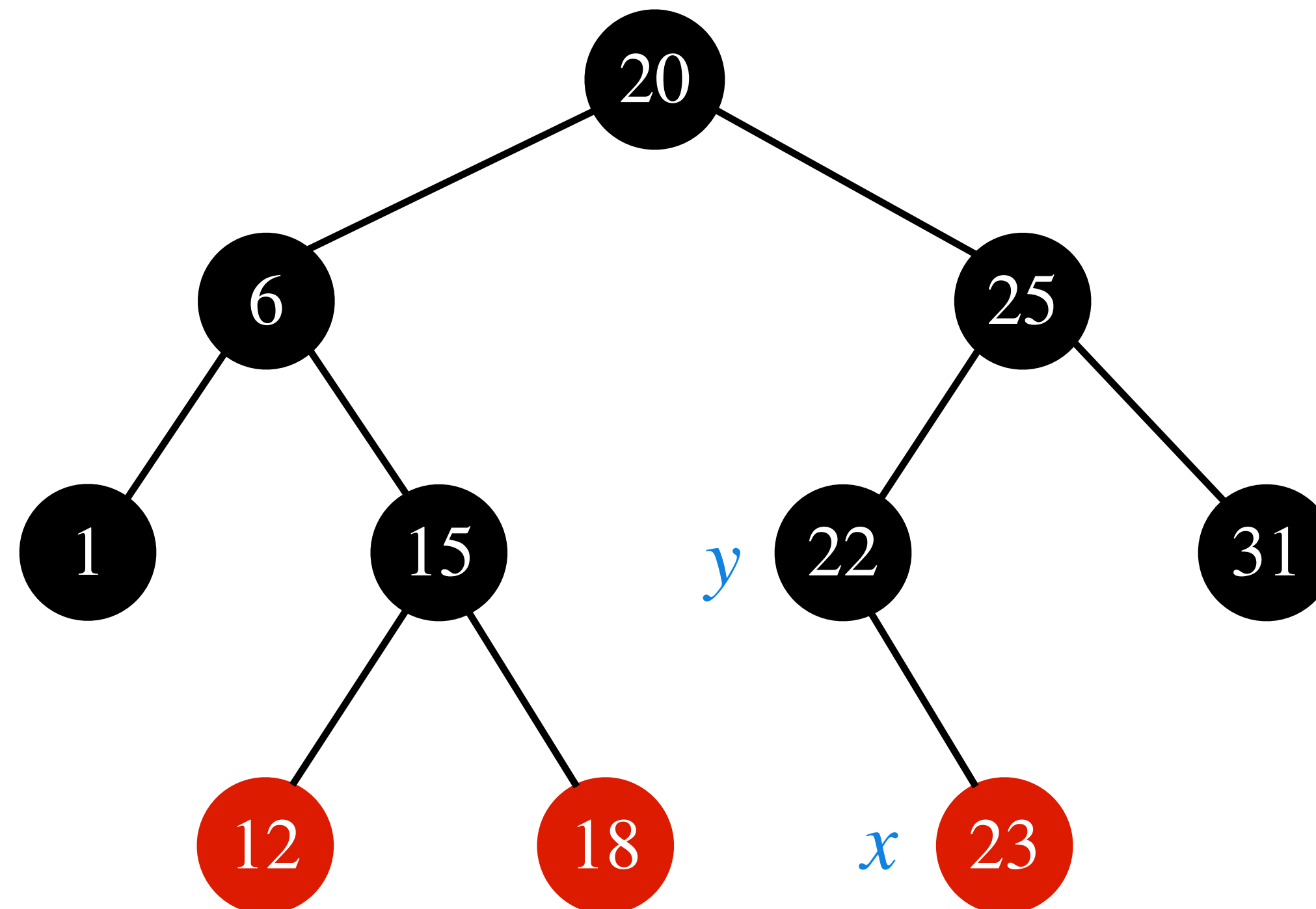
How can we fix the violation of property 4 and 5? Colour x black.

Fixing Violation of Only Property 5

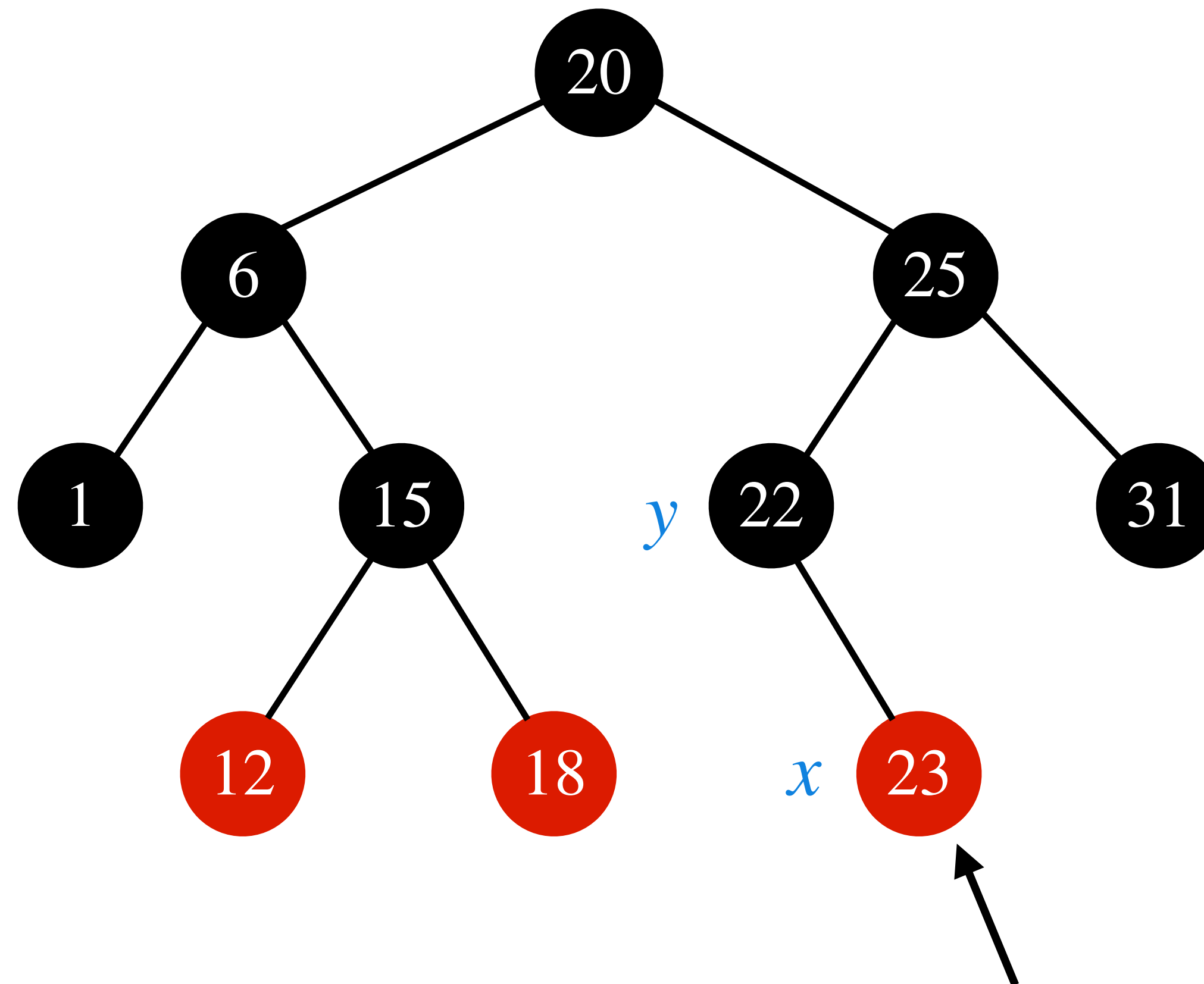
We need to find a fix up when **property 5** is the **only** property that is getting violated.

Fixing Violation of Only Property 5

Fixing Violation of Only Property 5

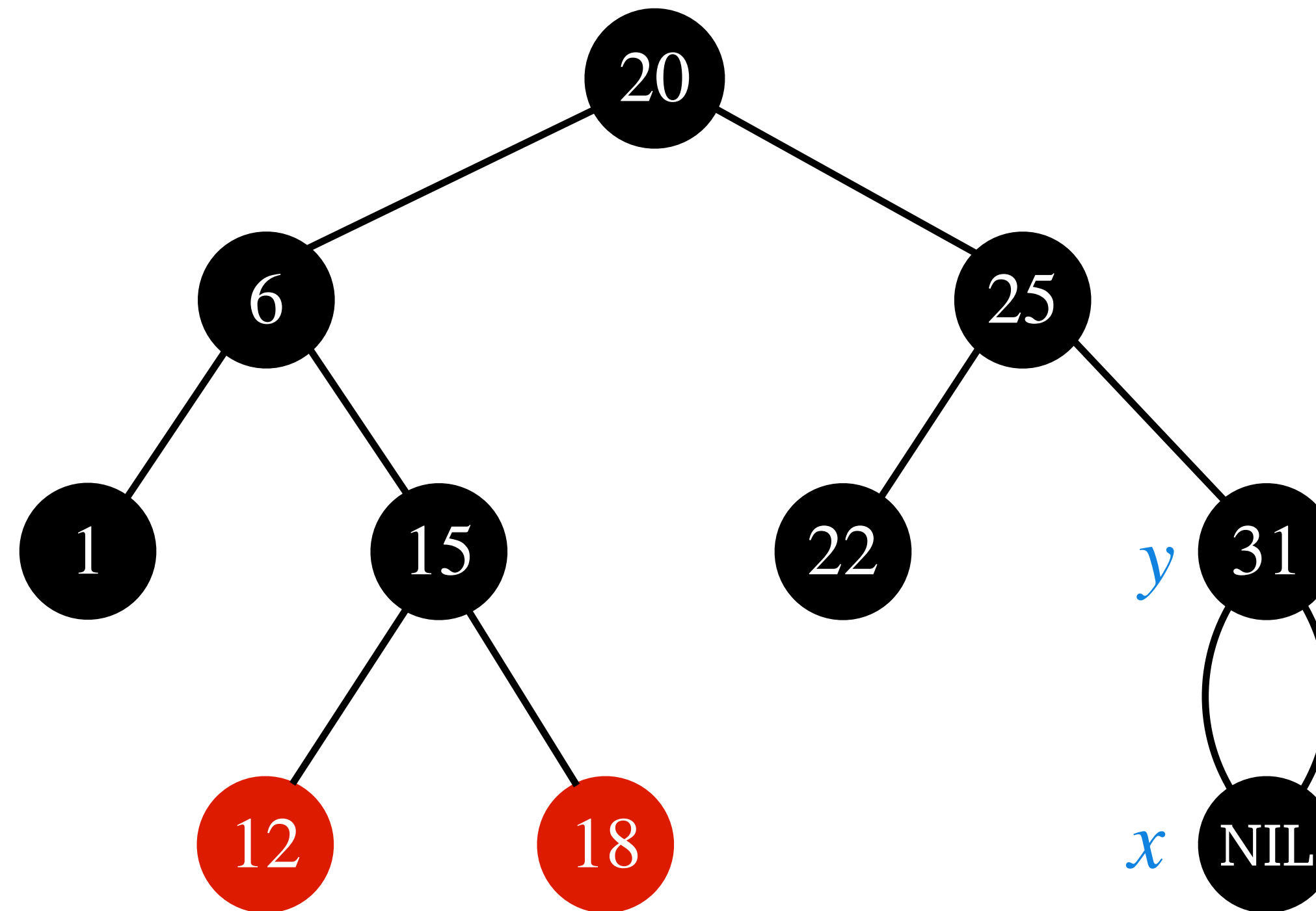


Fixing Violation of Only Property 5

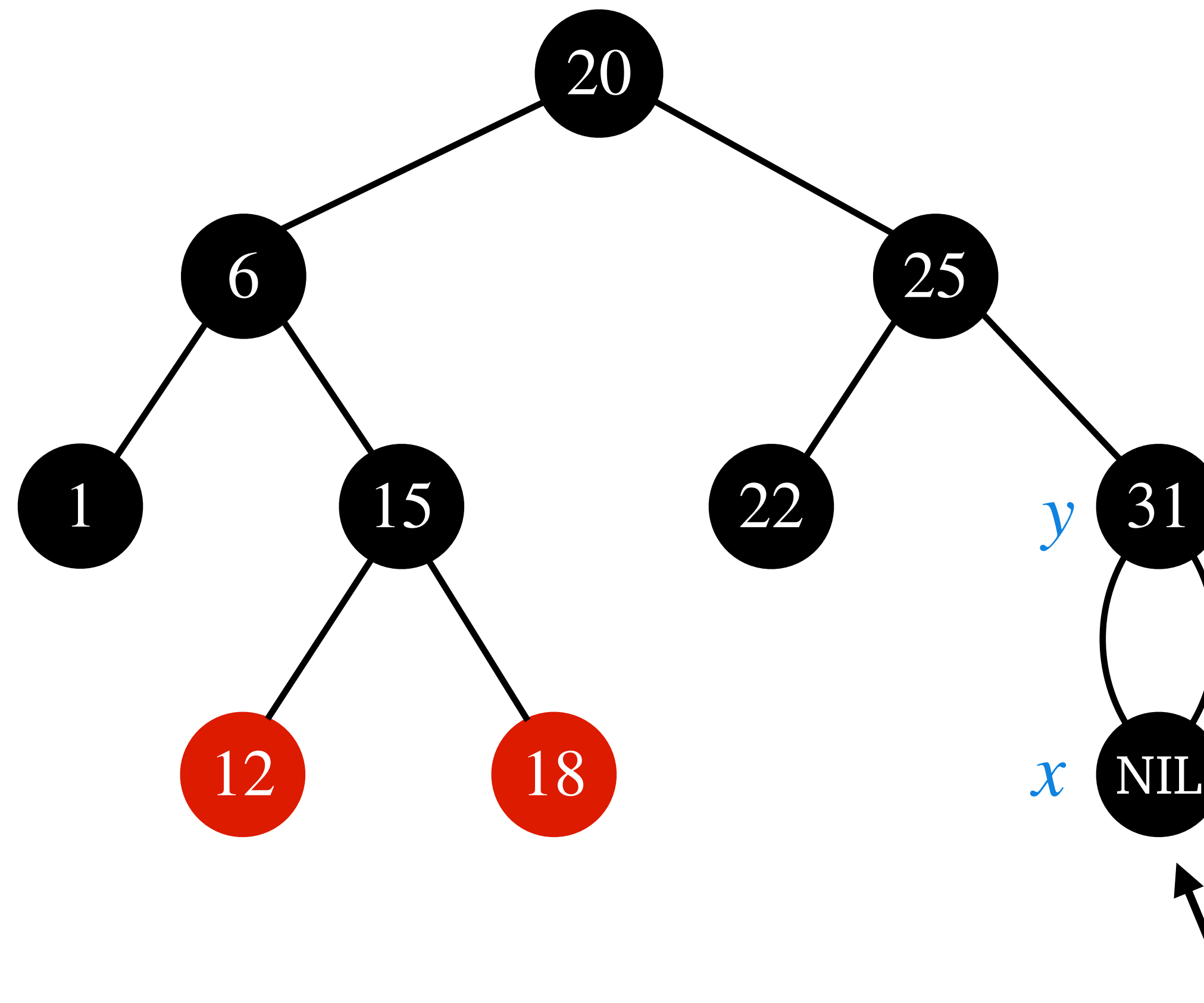


Fix up: Colour x black, when x is red

Fixing Violation of Only Property 5

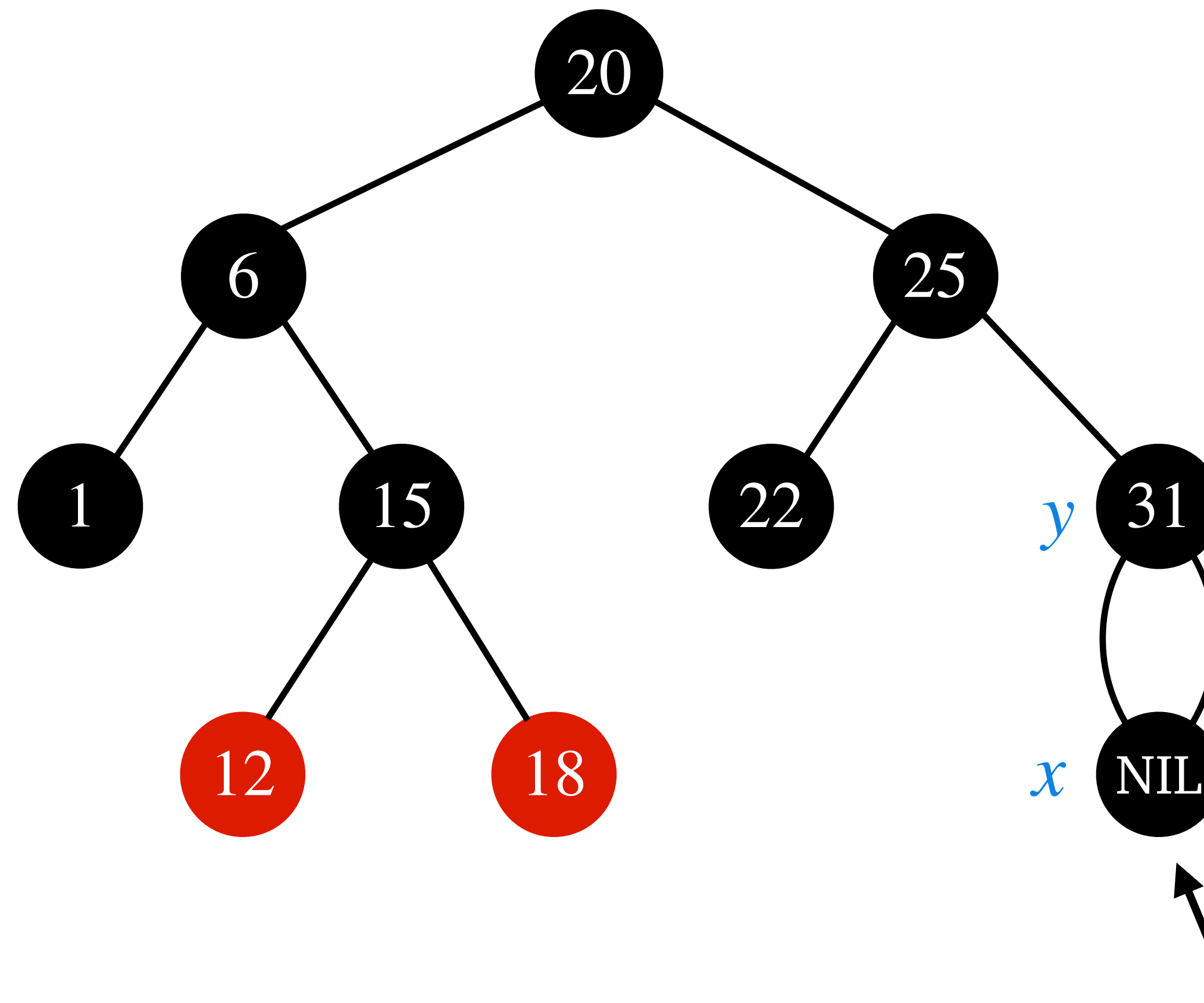


Fixing Violation of Only Property 5



Fix up: x is already black, now what?

Fixing Violation of Only Property 5



Fix up: Give x an extra black colour that it will lose without violating RB-tree properties.

Fixing Violation of Only Property 5

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is black:

- **Case 1:** x 's sibling w is red.

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
- **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.


Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
- **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
- **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.

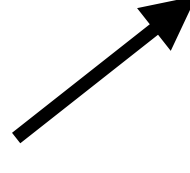
Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.  Case 1 will convert to Case 2, 3, or 4
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
- **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
- **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.

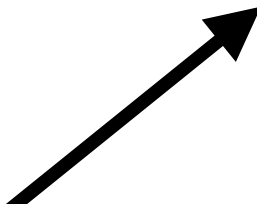
Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
 - **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
 - **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
 - **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.
- Case 2 will either terminate or
move x towards the root
- 

Fixing Violation of Only Property 5

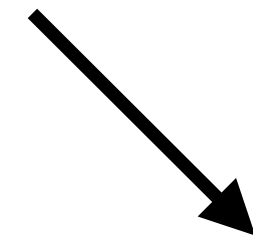
There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
 - **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
 - **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
 - **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.
- Case 3 will convert to Case 4
- 

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
- **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
- **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.



Case 4 will terminate

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is **black**:

- **Case 1:** x 's sibling w is **red**.
- **Case 2:** x 's sibling w is **black**, and both of w 's children are **black**.
- **Case 3:** x 's sibling w is **black**, and w 's left child is **red**, and w 's right child is **black**.
- **Case 4:** x 's sibling w is **black**, and w 's right child is **red**.

Note: We assume that x is non-NIL for the sake of simplicity.

Fixing Violation of Only Property 5

There are four cases when deletion causes only **Property 5's** violation and x is black:

- **Case 1:** x 's sibling w is red.
- **Case 2:** x 's sibling w is black, and both of w 's children are black.
- **Case 3:** x 's sibling w is black, and w 's left child is red, and w 's right child is black.
- **Case 4:** x 's sibling w is black, and w 's right child is red.

Idea: Give x an extra black colour that it will lose without violating RB-tree properties.