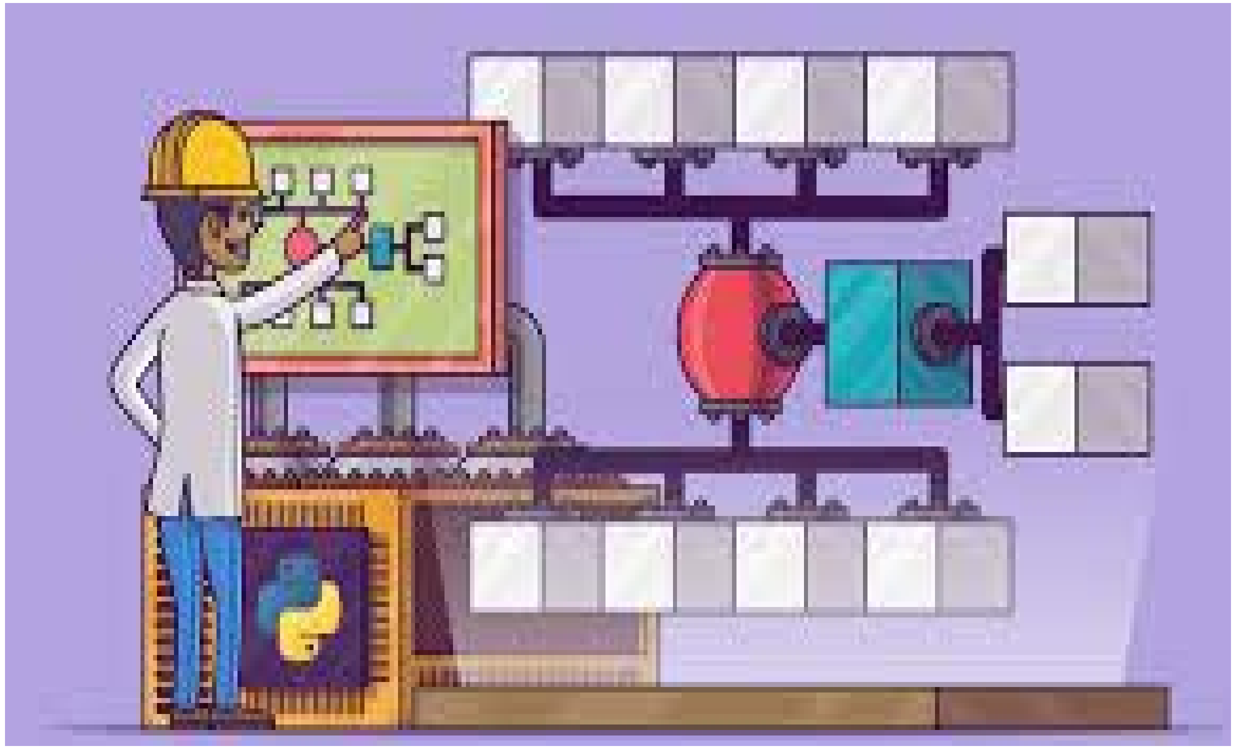
ساختمان داده ها

مدرس:
سمانه حسینی سمنانی

دانشگاه صنعتی اصفهان- دانشکده برق و کامپیوتر



درخت ها

- مفاهیم اولیه
- پیمایش درخت
- درخت دودویی معادل
- پیاده سازی درخت
- درخت جستجوی دودویی
- درخت عبارت
- Heap tree (هرم بیشینه)

درخت ها

- Red-black tree

Red-black tree

Red-black tree

- binary search tree
  - SEARCH,
  - PREDECESSOR,
  - SUCCESSOR,
  - MINIMUM,
  - MAXIMUM,
  - INSERT,
  - DELETE

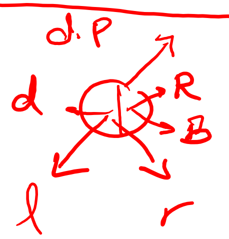
$O(h)$

ارتفاع

Red-black tree

- Red-black trees are one of many search-tree schemes that are “balanced” in order to guarantee that basic dynamic-set operations take O(log n) time in the worst case.

- A ***red-black tree*** is a binary search tree with one extra bit of storage per node: its ***color***, which can be either RED or BLACK.



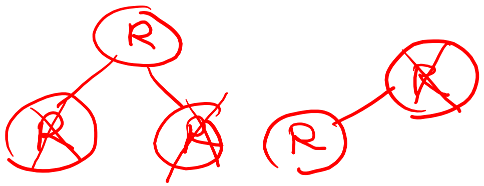
Red-black tree

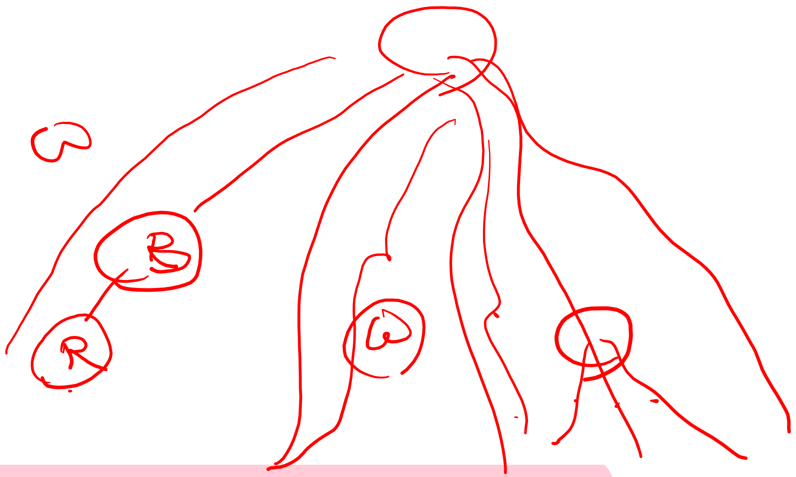
- Each node of the tree now contains the attributes color, key, left, right, and p.
- If a child or the parent of a node does not exist, the corresponding pointer attribute of the node contains the value NIL.

Red-black tree

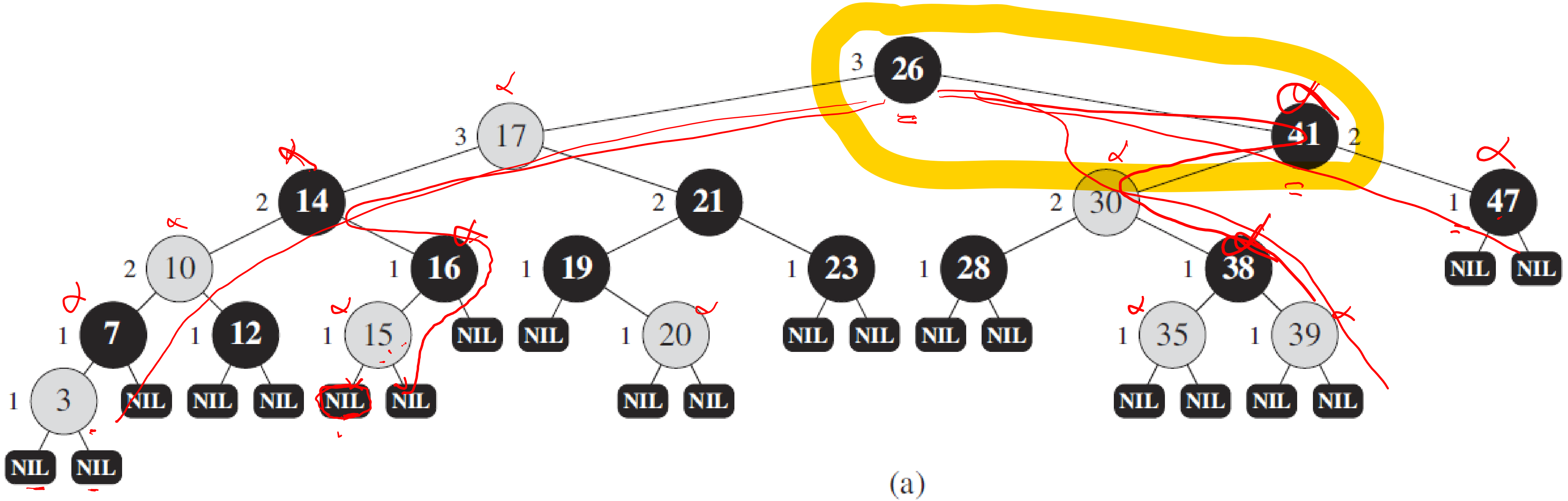
- A red-black tree is a binary tree that satisfies the following red-black properties:

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





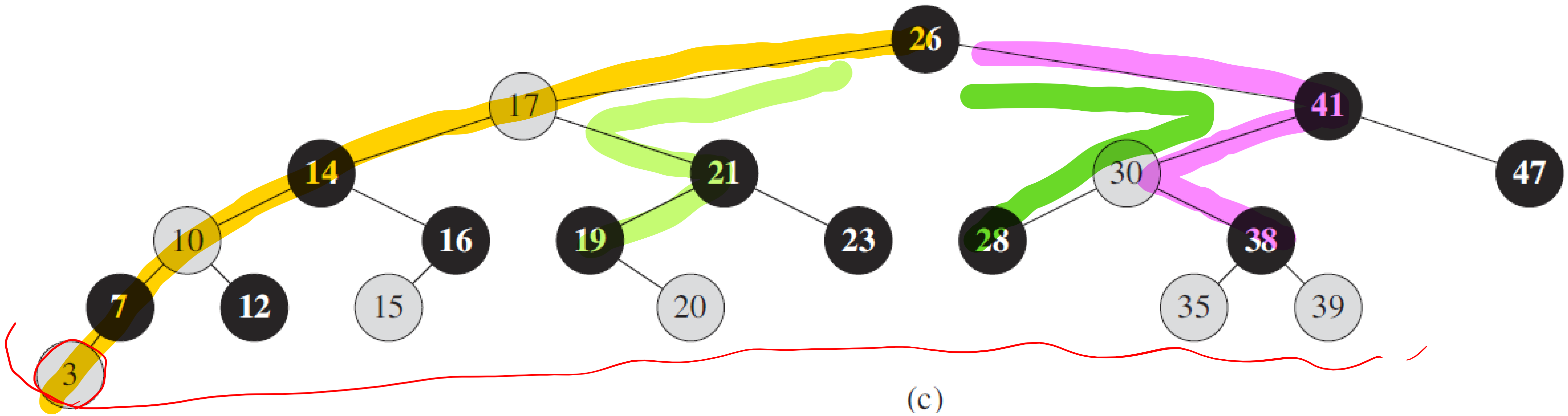
Red-black tree



(a)

سمانه حسینی سمنانی
هیات علمی دانشکده برق و کامپیوتر- دانشگاه صنعتی اصفهان

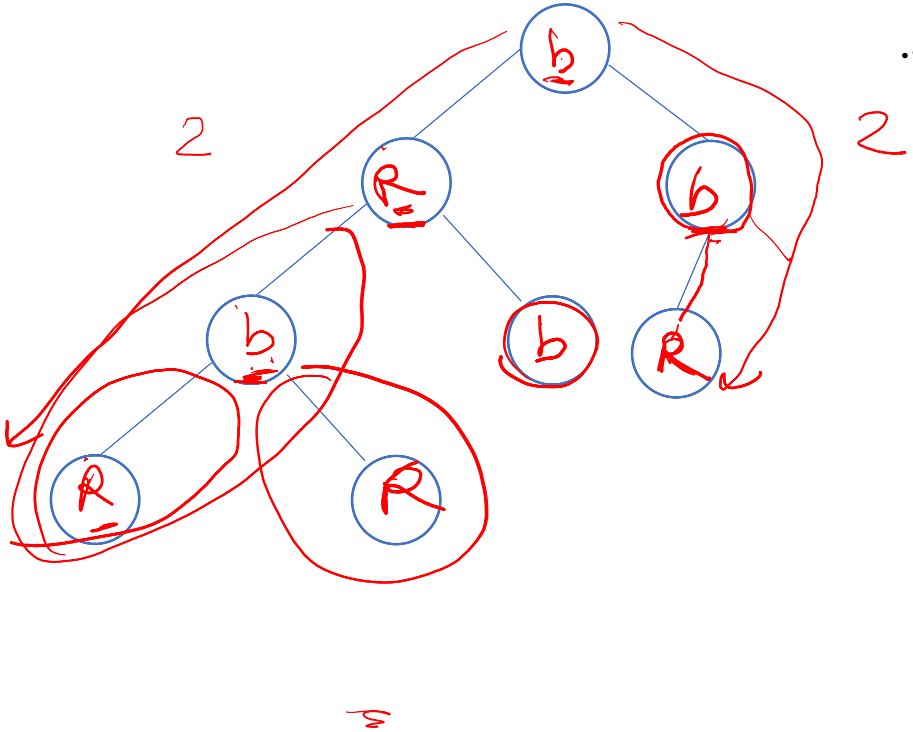
Red-black tree

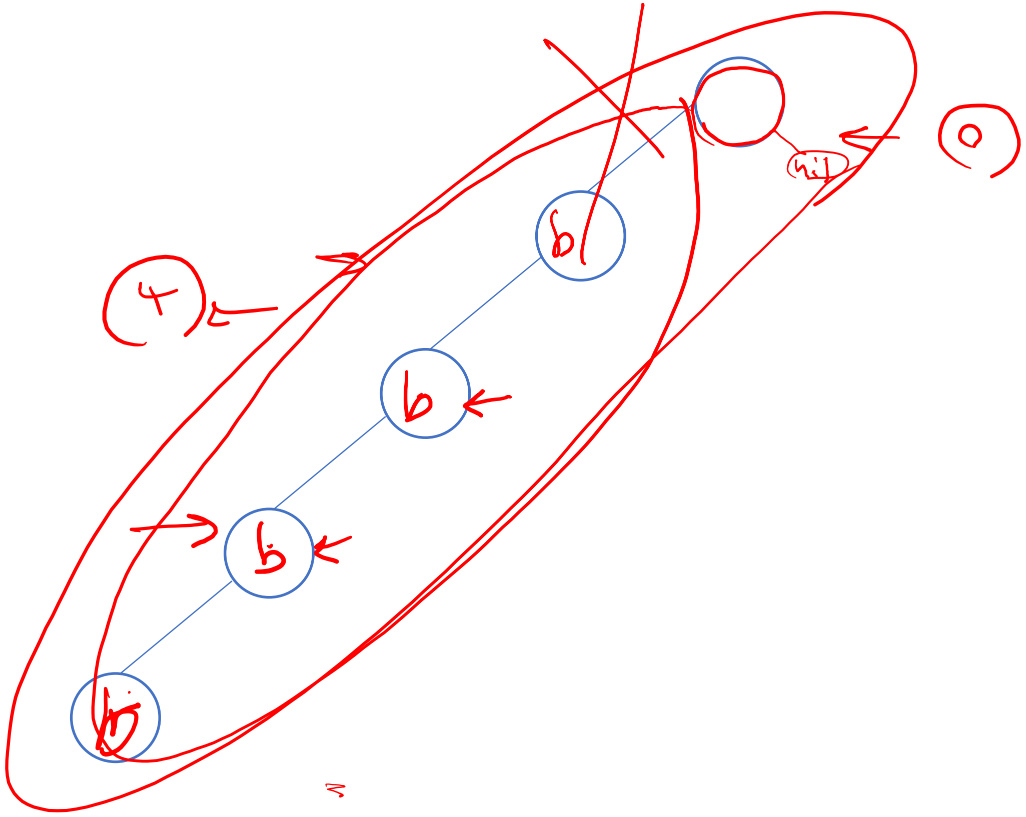


(c)

Red-black tree

- درخت های روبرو را طوری رنگ آمیزی کنید که red-black باشد.





Red-black tree

- By constraining the node colors on any simple path from the root to a leaf, red-black trees ensure that no such path is more than twice as long as any other, so that the tree is approximately **balanced**.

Lemma 13.1

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

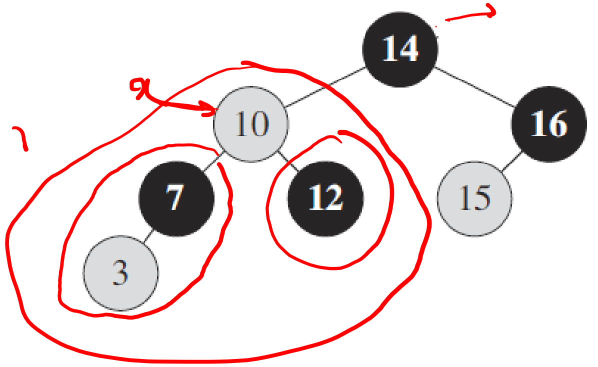
Red-black tree

Lemma 13.1

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

- $bh(x)$: تعداد نودهای سیاه از نود x تا برگ به غیر از خود نود

the number of black nodes on any simple path from, but not including, a node x down to a leaf the ***black-height*** of the node, denoted bh(x)



bh(x) = ۱

bh(x) = ۱

subtree rooted at any node x contains at least $2^{bh(x)} - 1$ internal nodes.

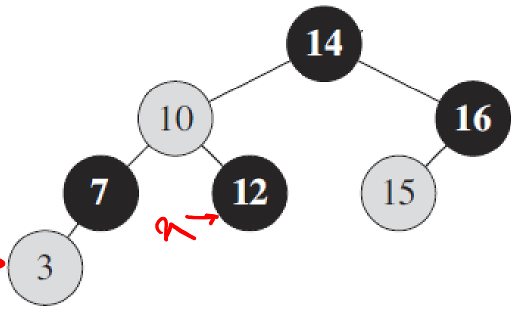
$2^1 - ۱ = ۱$

- اثبات استقرایی روی ارتفاع x

Red-black tree

subtree rooted at any node x contains at least $2^{\mathrm{bh}(x)} - 1$ internal nodes.

- if the height of x is 0, then x must be a leaf (T.nil),
- the subtree rooted at x indeed contains at least $2^{\mathrm{bh}(x)} - 1 = 2^0 - 1 = 0$ internal nodes



bh(x)

x

- Suppose: $bh(x) = k \;\; \rightarrow number\ of\ nodes\ in\ the\ subtree \geq 2^{k} - 1$

bh(x) = 0

- prove: $bh(x) = k + 1 \;\; \rightarrow number\ of\ nodes\ in\ the\ subtree \geq 2^{k+1} - 1$

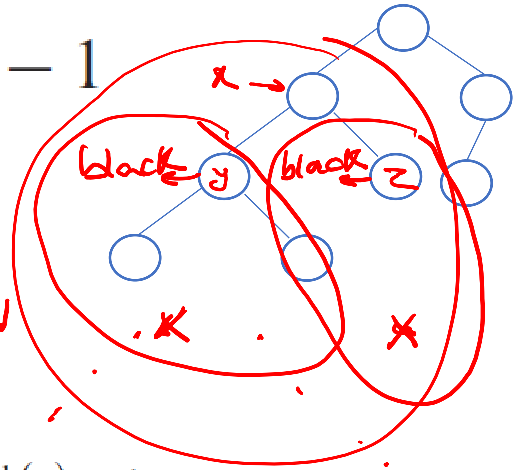
Red-black tree

subtree rooted at any node x contains at least $2^{\mathrm{bh}(x)} - 1$ internal nodes.

- Suppose: $bh(x) = k \;\; \rightarrow number\ of\ nodes\ in\ the\ subtree \geq 2^k - 1$
- prove: $bh(x) = k + 1 \;\; \rightarrow number\ of\ nodes\ in\ the\ subtree \geq 2^{k+1} - 1$
- Each child y and z has a black-height of either $\mathrm{bh}(x)$ or $\mathrm{bh}(x) - 1$
- depending on whether its color is red or black, respectively.
- y and z has at least $2^{\mathrm{bh}(x)-1} - 1$ internal nodes.
- The subtree rooted at x has at least $(2^{\mathrm{bh}(x)-1} - 1) + (2^{\mathrm{bh}(x)-1} - \cancel{1}) + \cancel{1} = 2^{\mathrm{bh}(x)} - 1$ internal nodes.

$bh(x) = k+1$

$bh(y)$, $bh(z)$ $\rightarrow bh(x)$, $bh(x) - 1$



Red-black tree

Lemma 13.1

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

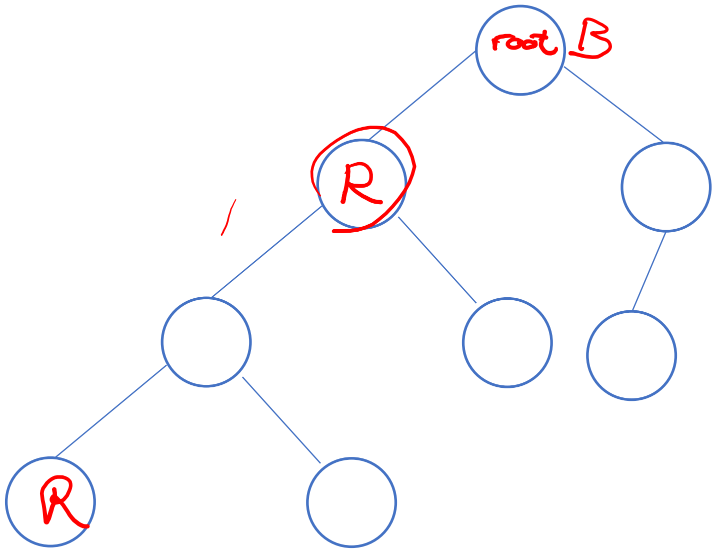
- $bh(root) \geq \frac{h}{2}$

subtree rooted at any node x contains at least $2^{bh(x)} - 1$ internal nodes.

- تعداد گره ها $= n \geq 2^{bh(root)} - 1 \geq 2^{\frac{h}{2}} - 1$

- $n + 1 \geq 2^{\frac{h}{2}} \quad \rightarrow h \leq 2\log(n+1)$

$\log(n+1) \geq \frac{h}{2} \log_2 2 \rightarrow h \leq 2\log(n+1)$



Red-black tree

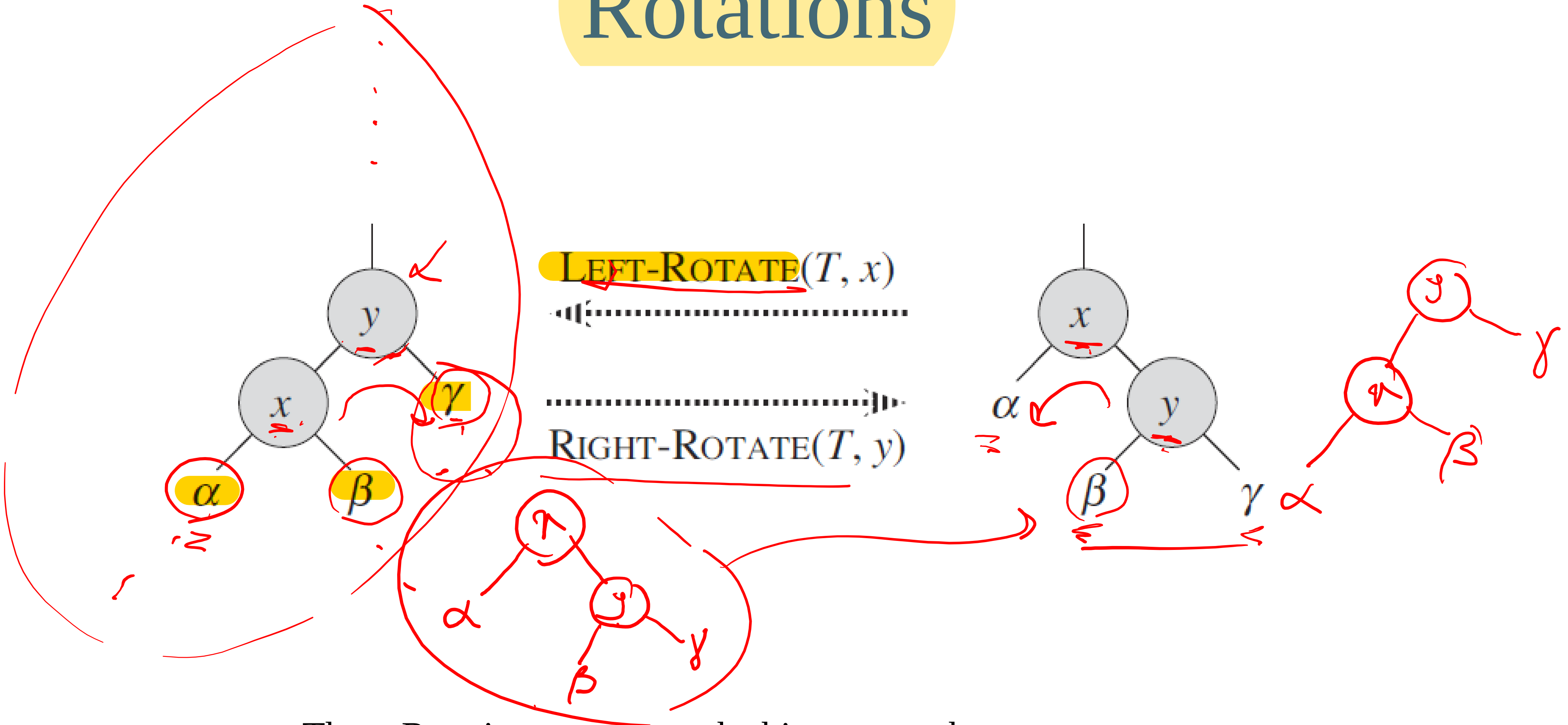
A red-black tree with n internal nodes has height at most $2 \lg(n + 1)$.

- SEARCH, MINIMUM, MAXIMUM, SUCCESSOR, and PREDECESSOR are O(log n)

- دنبال حفظ توازن درخت هستیم ← باید خاصیت قرمز- سیاه را در درج و حذف حفظ کنیم.

- INSERT and DELETE

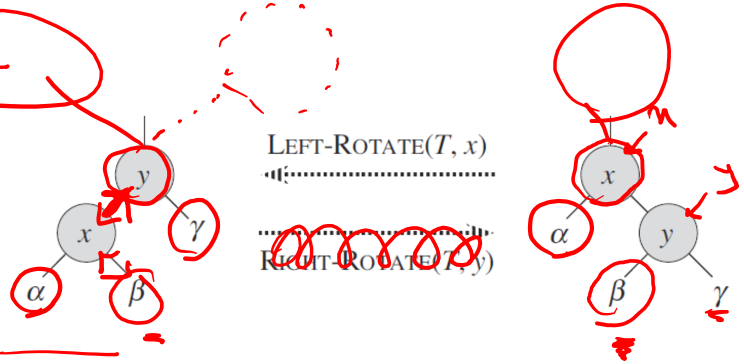
Rotations



These Rotations preserves the binary-search-tree property.

Rotations

```
LEFT-ROTATE(T, x)
    y = x.right                 // set y
    x.right = y.left            // turn y's left subtree into x's right subtree
    if y.left ≠ T.nil
        y.left.p = x
    y.p = x.p                   // link x's parent to y
    if x.p == T.nil
        T.root = y
    elseif x == x.p.left
        x.p.left = y
    else x.p.right = y
    y.left = x                  // put x on y's left
    x.p = y
```

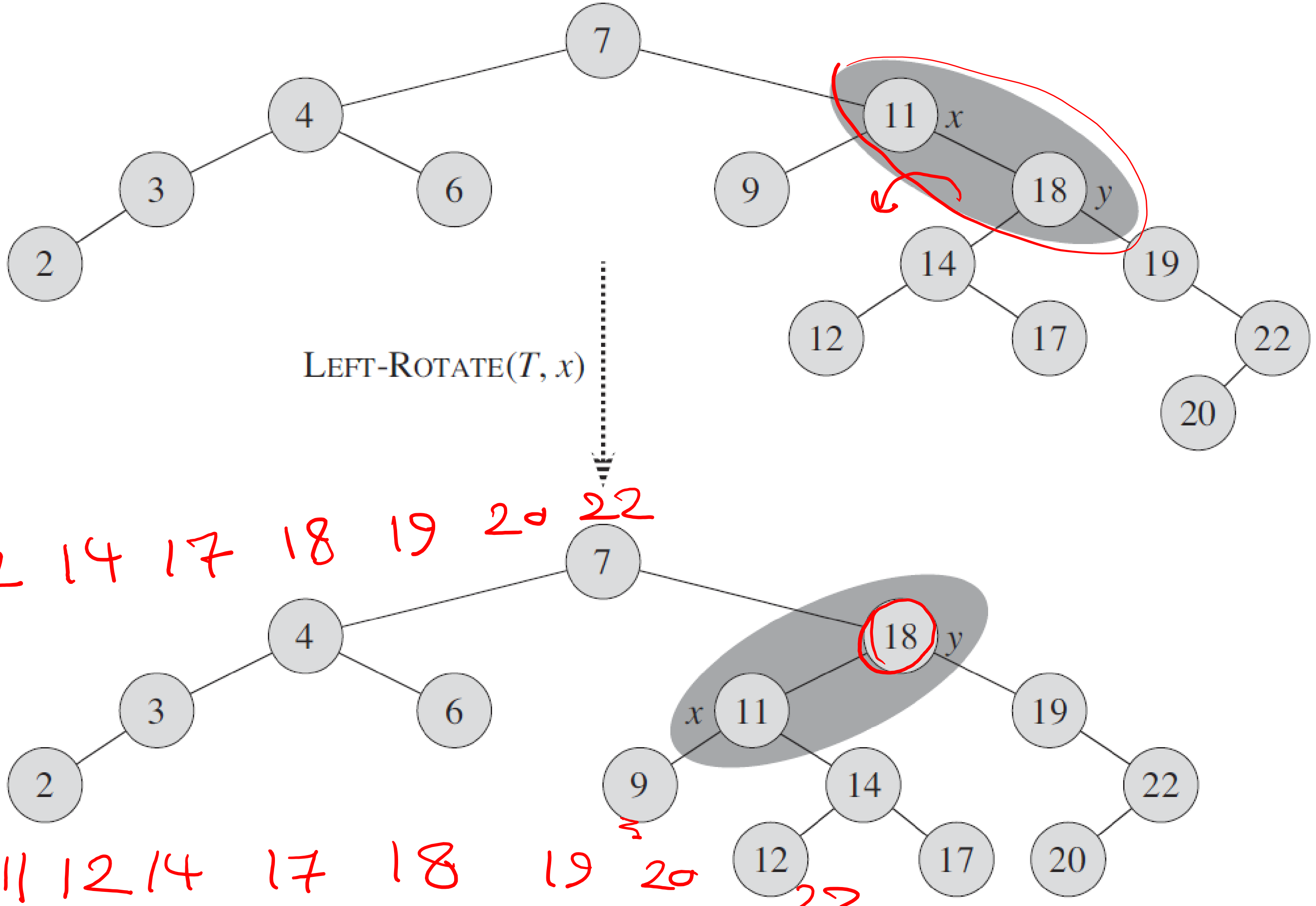


$\alpha\ x\ \beta\ y\ \gamma$ $\qquad$ $\alpha\ x\ \beta\ y\ \gamma$

$O(1)$

Rotations

In order tree walks of the input tree and the modified tree produce the same listing of key values.



2 3 4 6 7 9 11 12 14 17 18 19 20 22

2 3 4 6 7 9 11 12 14 17 18 19 20 22