case 3
case 1
RB-Insert (T,x)  
Tree-Insert(T,x)  
color[x]=red  

while $x \neq root[T]$ and color[p[x]]=red  
  if p[x]=left[p[p[x]]] then  
    y=right[p[p[x]]]  
    if color[y]=red then  
      \  
      \color[p[x]]=black  
      color[y]=black  
      color[p[p[x]]]=red  
    x=p[p[x]]  
  else (same as then clause, with “right” and “left” exchanged)  
  else (same as then clause, with “right” and “left” exchanged)
We need some auxiliary operations: Left-Rotate and Right-Rotate

```
Left-Rotate(T, x)
Right-Rotate(T, y)
```

```
x
  y
  γ
  x
  α
  β

Right-Rotate(T, y)
Left-Rotate(T, x)
```

```
x
  α
  β
  γ
  y
  β
  γ
```
Theorem: A red-black tree with \( n \) internal nodes has height at most \( 2 \lg (n+1) \)

Proof: Let \( h \) be the height of the tree.

--At least half the nodes on any path from root to a leaf must be black
--Then, \( bh(\text{root}) \) must be at least \( h/2 \)

--By Lemma 1, there are at least \( 2^{h/2} - 1 \) internal nodes in this tree
--Then, \( 2^{h/2} - 1 \leq n \)

\[
2^{h/2} \leq n + 1
\]

\[
h/2 \leq \lg (n + 1)
\]

\[
h \leq 2\lg (n + 1)
\]

QED

In other words, the height of a red-black tree with \( n \) nodes is \( O(\lg n) \)
Lemma 1: A subtree rooted at any node $x$ contains at least $2^{bh(x)} - 1$ internal nodes

Proof: By induction on the height of $x$.

Basis: $x$ is height 0. Then $x$ is a leaf (NIL) and the subtree rooted at $x$ contains at least $2^{bh(x)} - 1 = 2^0 - 1 = 0$ internal nodes

Inductive hypothesis:
If the Lemma holds for every node of height 1, 2, ..., $k-1$, then the Lemma holds for any node of height $k$

Proof of Inductive hypothesis:
-- consider a node $x$ at height $= k$; since $k > 0$, $x$ must have 2 children
-- each child has a black height that is either $bh(x)$ or $bh(x) - 1$
-- height of each child is $< k$
-- therefore by the assumption of the inductive hypothesis, each child has at least $2^{bh(x) - 1} - 1$ internal nodes in its subtree
-- therefore, $x$ has at least
\[ \left( 2^{bh(x) - 1} - 1 \right) + \left( 2^{bh(x) - 1} - 1 \right) + 1 = 2^{bh(x)} - 1 \]
internal nodes in its subtree
QED
Proofs by induction

Lemma: The sum $1+2+3+4+5+\ldots+n=\frac{n(n+1)}{2}$

Proof: Basis: prove the Lemma for $n=1$

$$\frac{1(1+1)}{2}=1$$

Inductive hypothesis: If the Lemma holds for every value $n=1,2,3,\ldots,k-1$, then the Lemma holds for value $k$

$$\left(\text{If for every } k-1 \geq 1, 1+2+3+4+5+\ldots+k-1=\frac{(k-1)(k-1+1)}{2},\text{ then } 1+2+3+4+5+\ldots+k=\frac{k(k+1)}{2}\right)$$

Proof of Inductive hypothesis:

$$1+2+3+4+5+\ldots+k=(1+2+3+4+5+\ldots+k-1)+k=$$

$$=\frac{(k-1)(k-1+1)}{2}+k=\frac{(k-1)k}{2}+\frac{2k}{2}=\frac{(k-1+2)k}{2}=\frac{k(k+1)}{2}$$

QED
Red-black tree: binary search tree with 1 extra bit per node -- it’s color must satisfy red-black properties:

1. every node is either red or black
2. every leaf (NIL) is black
3. if a node is red, then both its children are black
4. every simple path from a node to a descendant leaf contains the same number of black nodes

leaves (nil) are called external nodes
non-leaf nodes are called internal nodes

black height of node x = bh(x) = the number of black nodes on any path from x to a leaf (but not counting node x)
Lecture 10: Red-Black Trees

Binary Search trees:
    all query operations run in $O(h)$ time, where $h=$height of tree

How to insure that $h$ is small?

If the tree is completely balanced, then $h = \lceil \lg n \rceil$

If we want to keep the tree completely balanced, it could take us as much as $\Omega(n)$ to do an Insert or a Delete operation

Our objective:
    keep it balanced enough so that $h$ is at most roughly $2 \lg n$
    and Inserts and Deletes are still doable in $O(h)$ time