
Algorithms and Data Structures

CSCI 4041

Session 11

Tree Insert

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

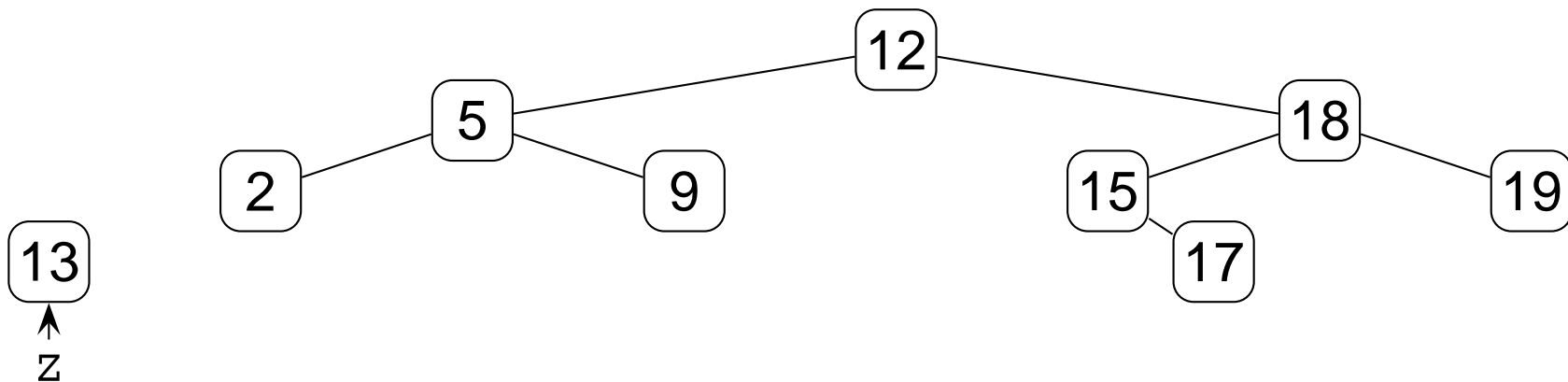
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (1)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

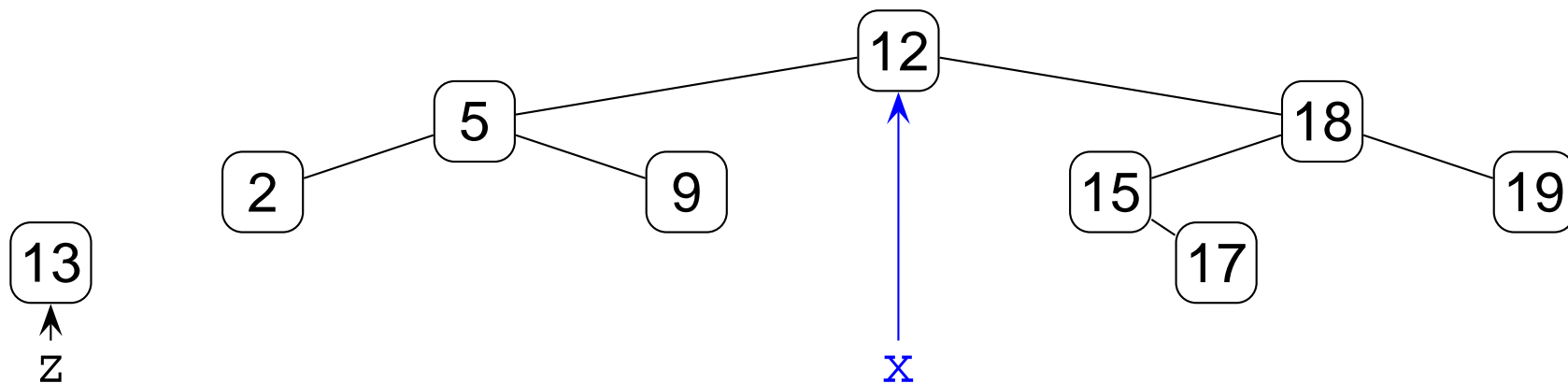
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (2)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

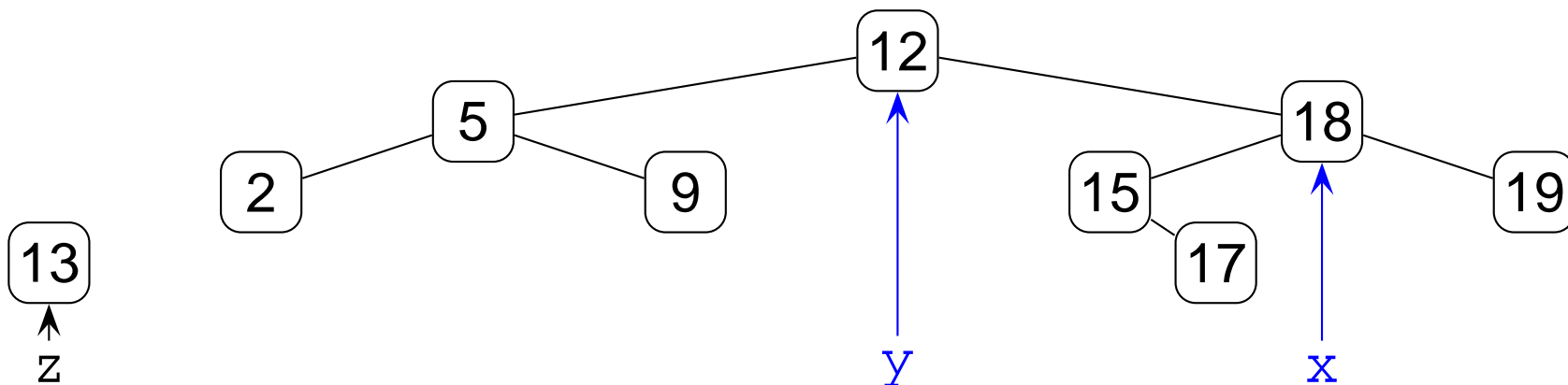
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (3)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

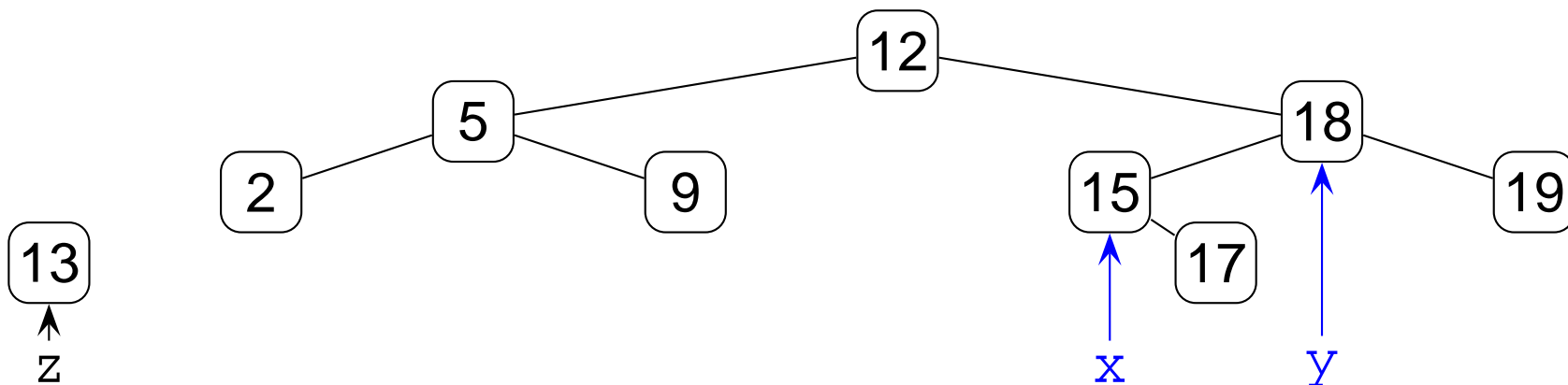
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (4)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

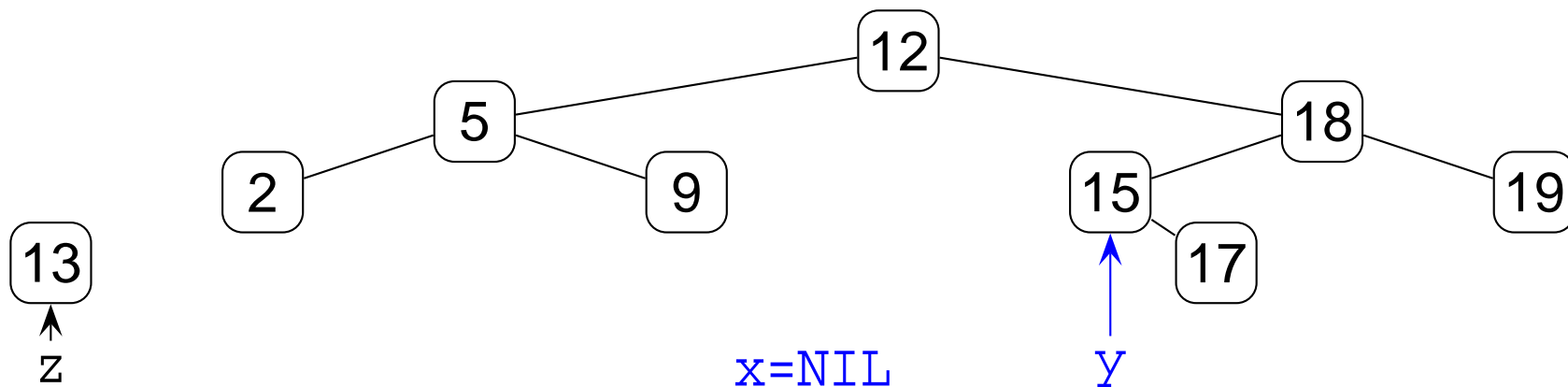
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (5)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

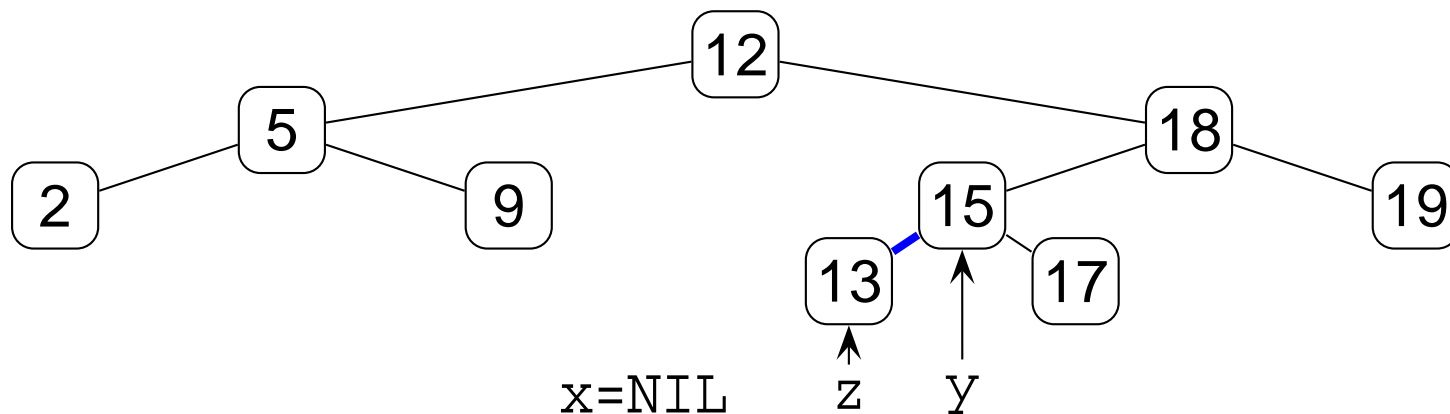
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Insert (done)

treeInsert(Tree T, Node z):

loop from $y \leftarrow \text{NIL}$, $x \leftarrow \text{root}(T)$ **while** $x \neq \text{NIL}$

$y \leftarrow x$

if $\text{key}(z) < \text{key}(x)$ **then** $x \leftarrow \text{left}(x)$

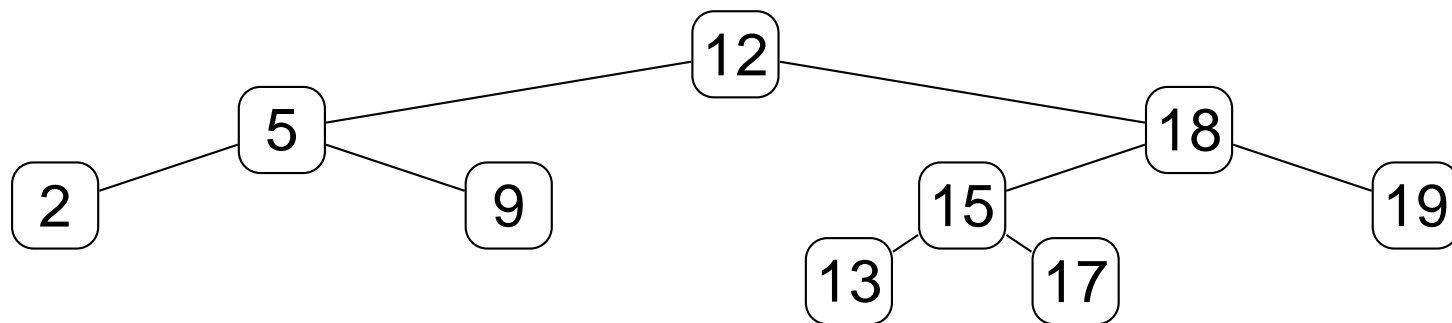
else $x \leftarrow \text{right}(x)$

$\text{parent}(z) \leftarrow y$

if $y = \text{NIL}$ **then** $\text{root}(T) \leftarrow z$

else if $\text{key}(z) < \text{key}(y)$ **then** $\text{left}(y) \leftarrow z$

else $\text{right}(y) \leftarrow z$



Tree Delete [target has no children]

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

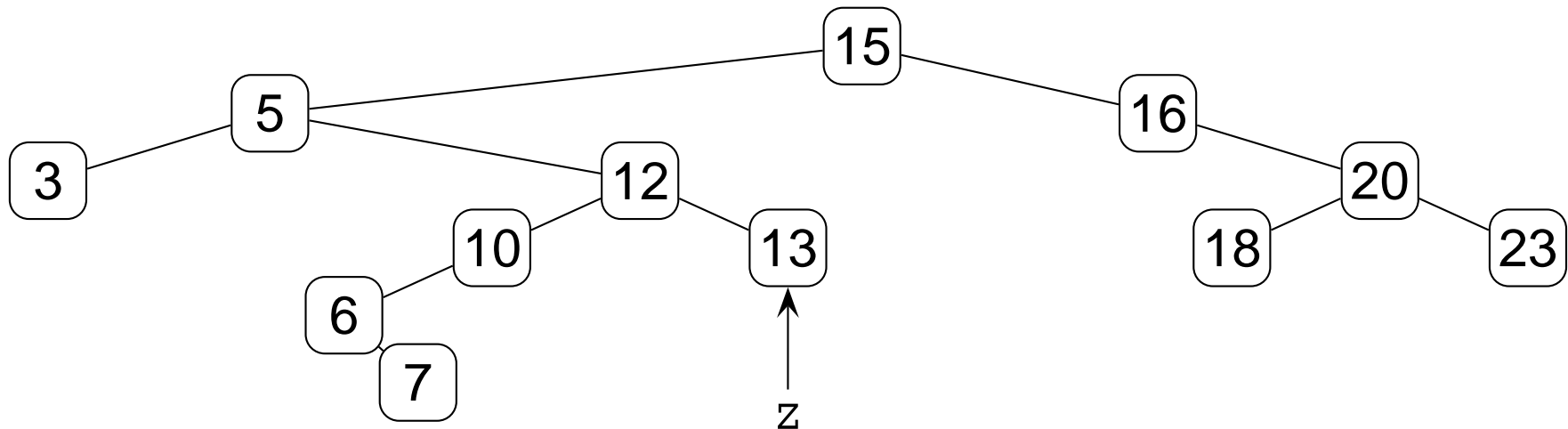
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has no children] (1)

treeDelete(Tree T, Node z):

if $\text{left}(z)=\text{NIL}$ or $\text{right}(z)=\text{NIL}$ **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

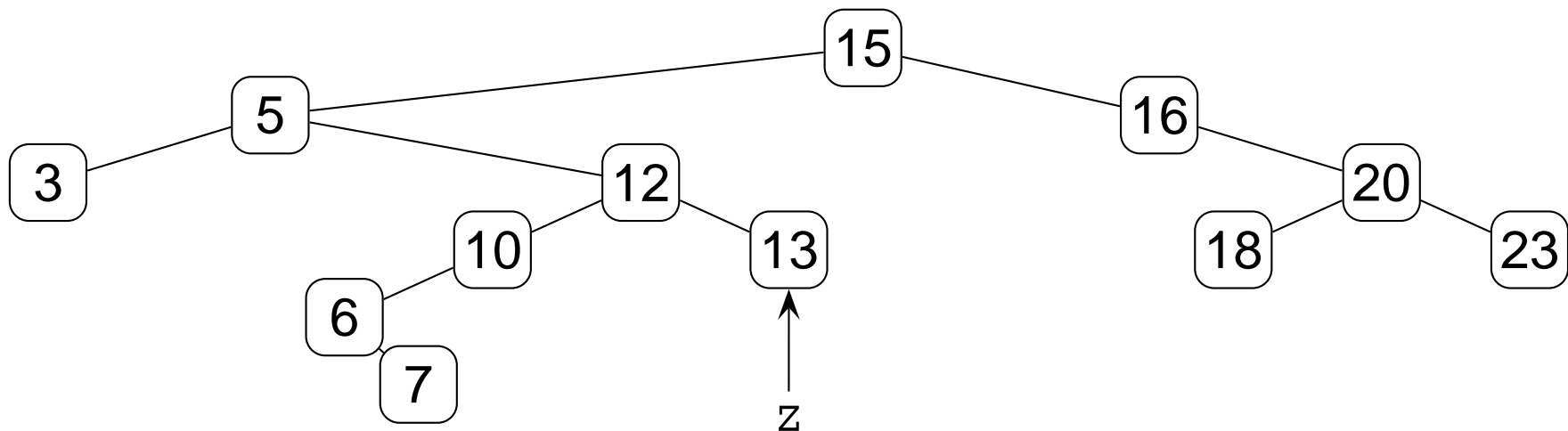
if $\text{left}(y) \neq \text{NIL}$ **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y)=\text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y=\text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$
else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



y $x=\text{NIL}$

Tree Delete [target has no children] (2)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

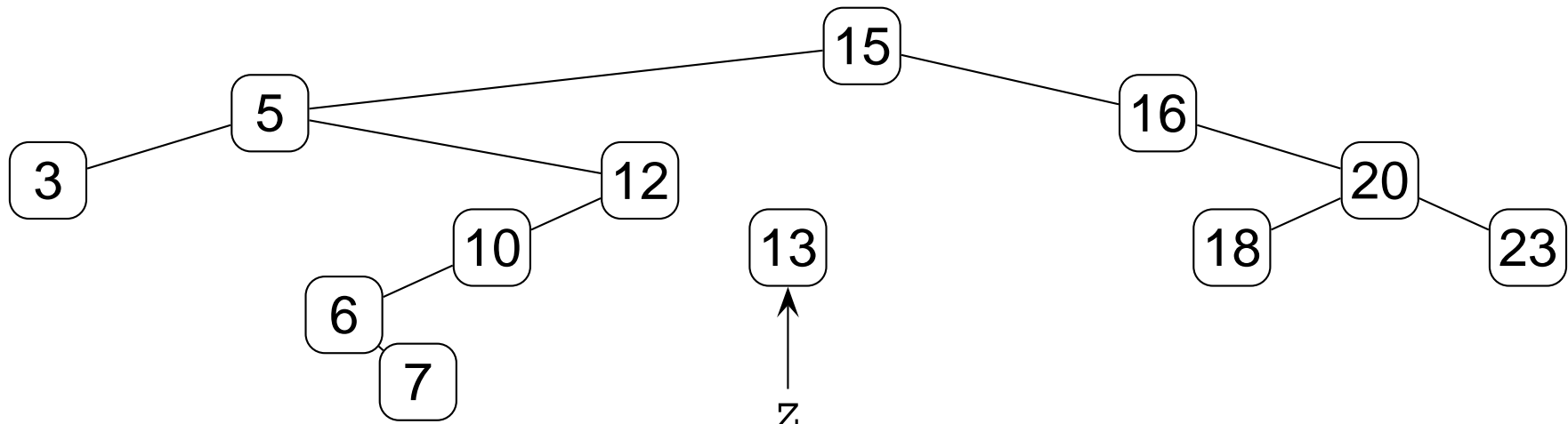
if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$
else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Y x=NIL

Tree Delete [target has no children] (done)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

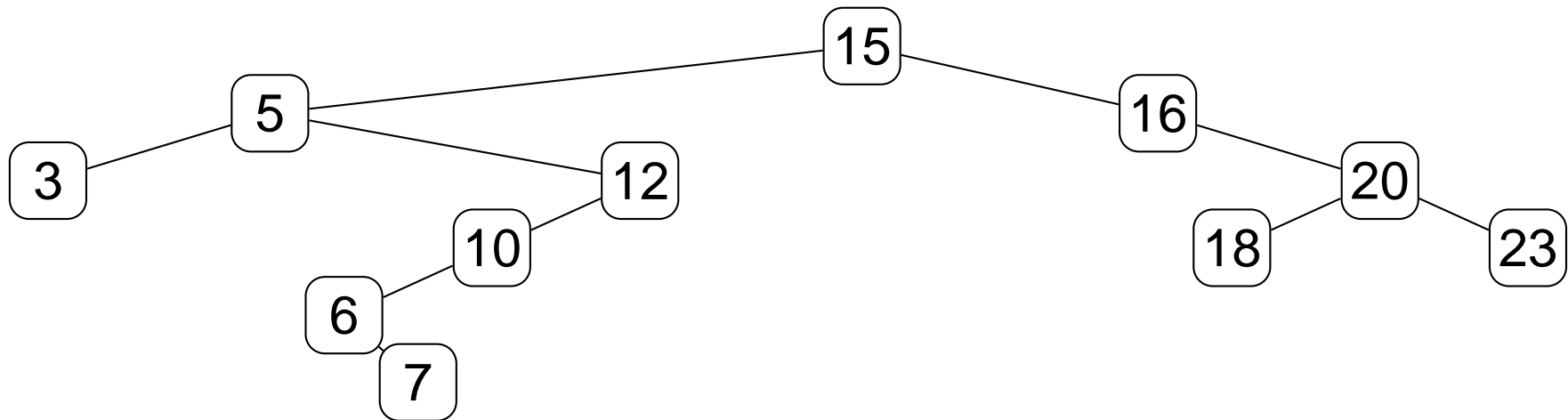
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has one child]

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

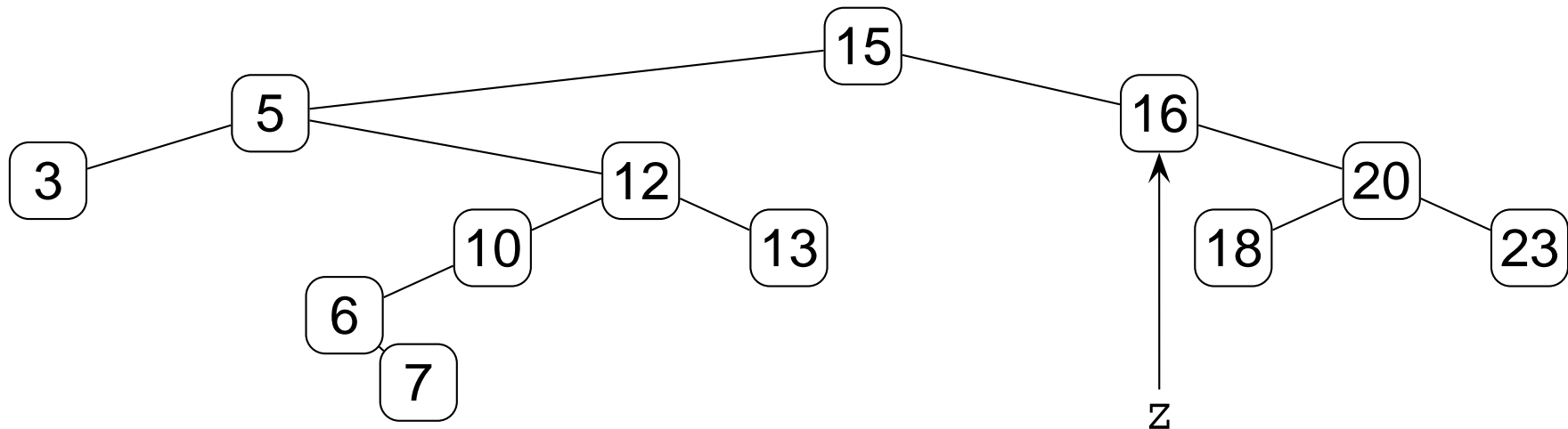
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has one child] (1)

treeDelete(Tree T, Node z):

if $\text{left}(z)=\text{NIL}$ or $\text{right}(z)=\text{NIL}$ **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if $\text{left}(y) \neq \text{NIL}$ **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

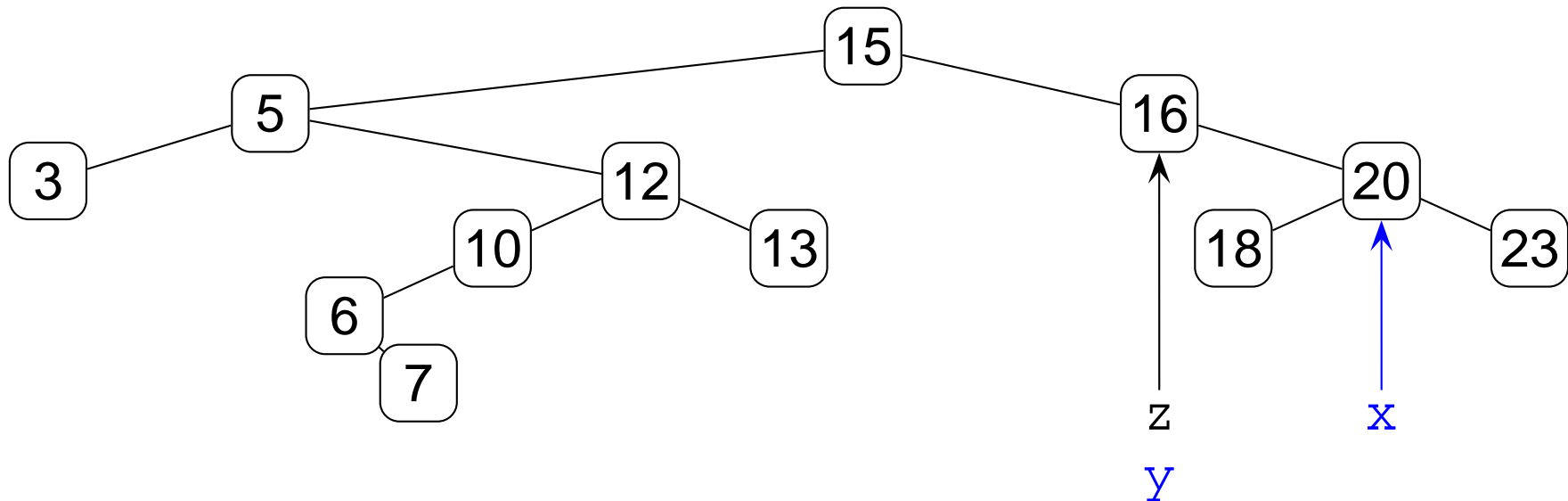
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y)=\text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y=\text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has one child] (2)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

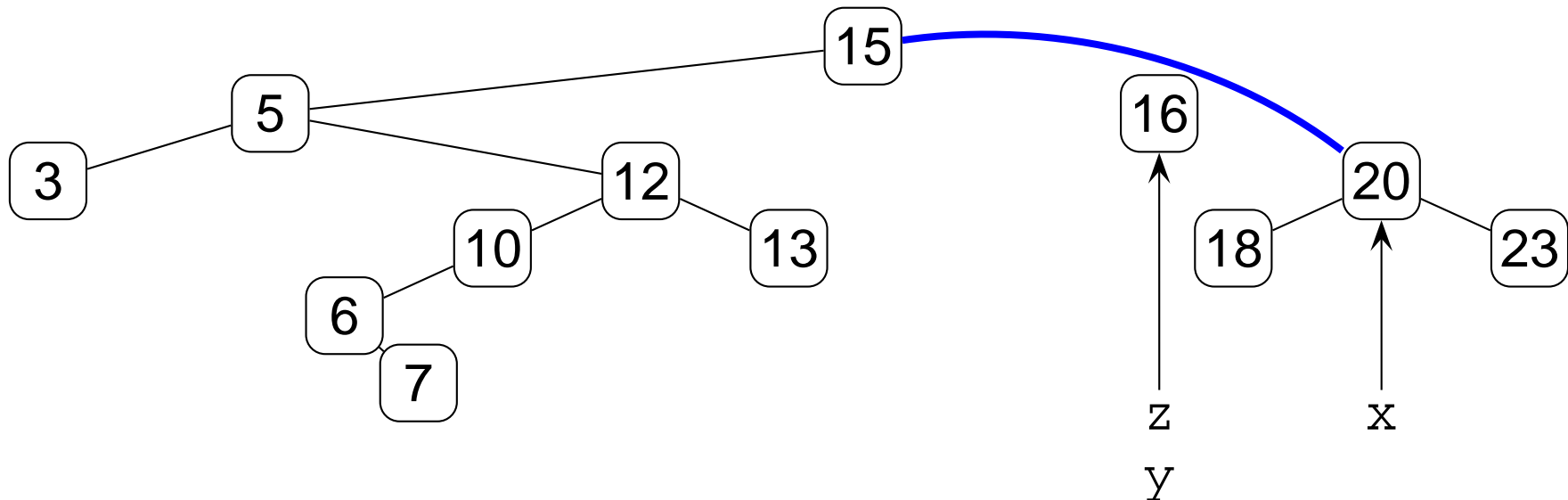
if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$
else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has one child] (done)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

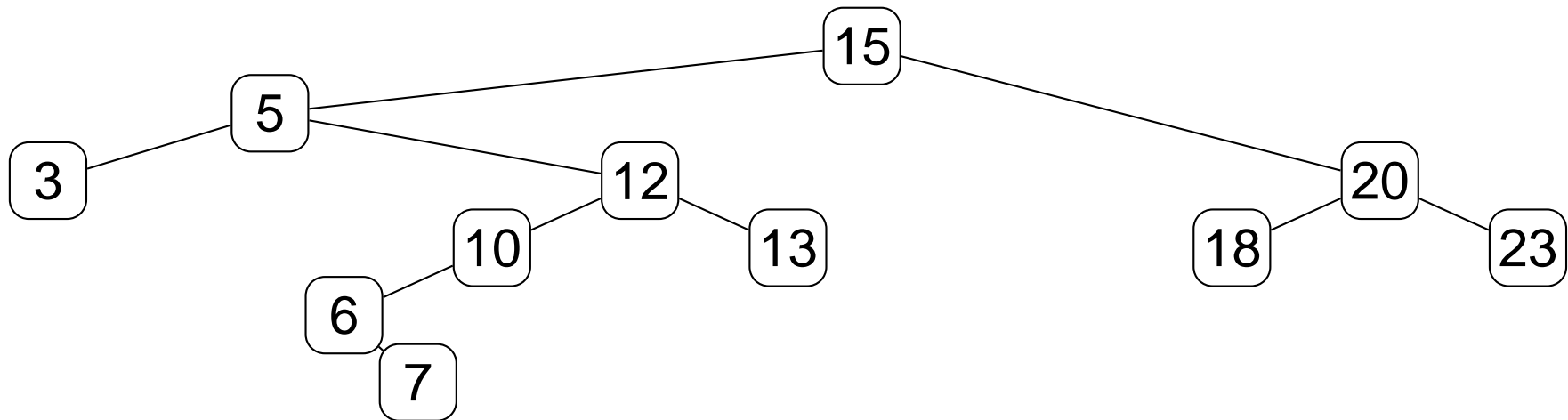
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has two children]

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

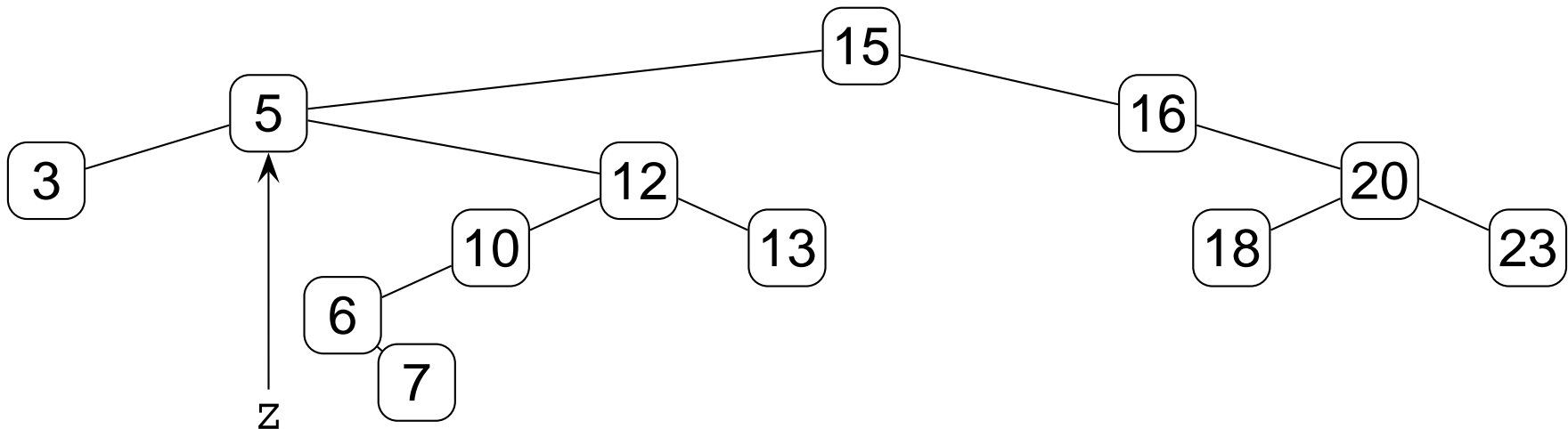
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has two children] (1)

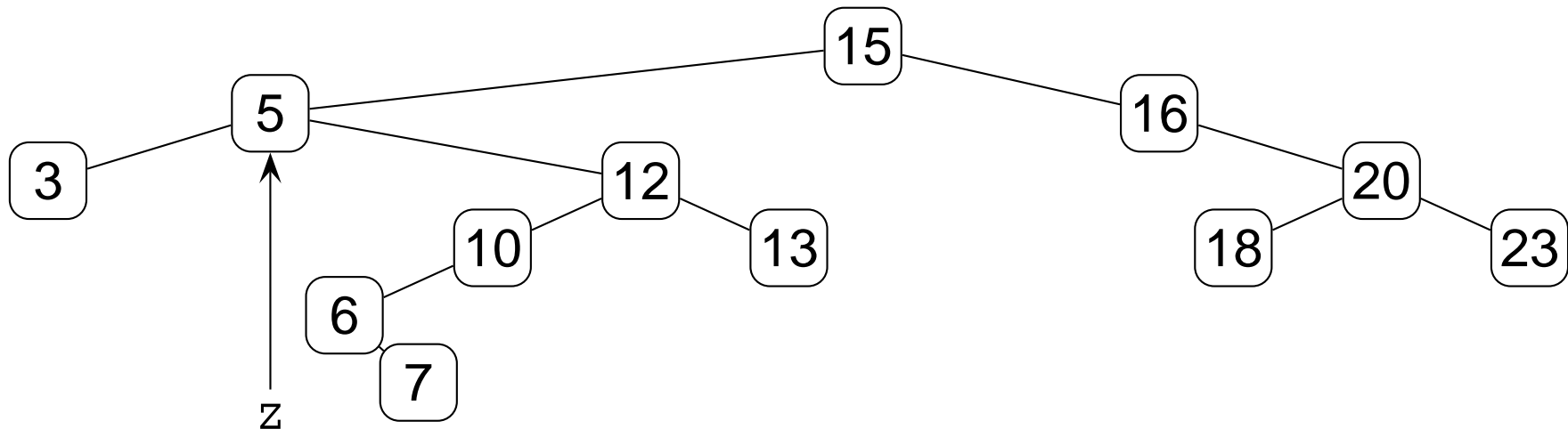
treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

...

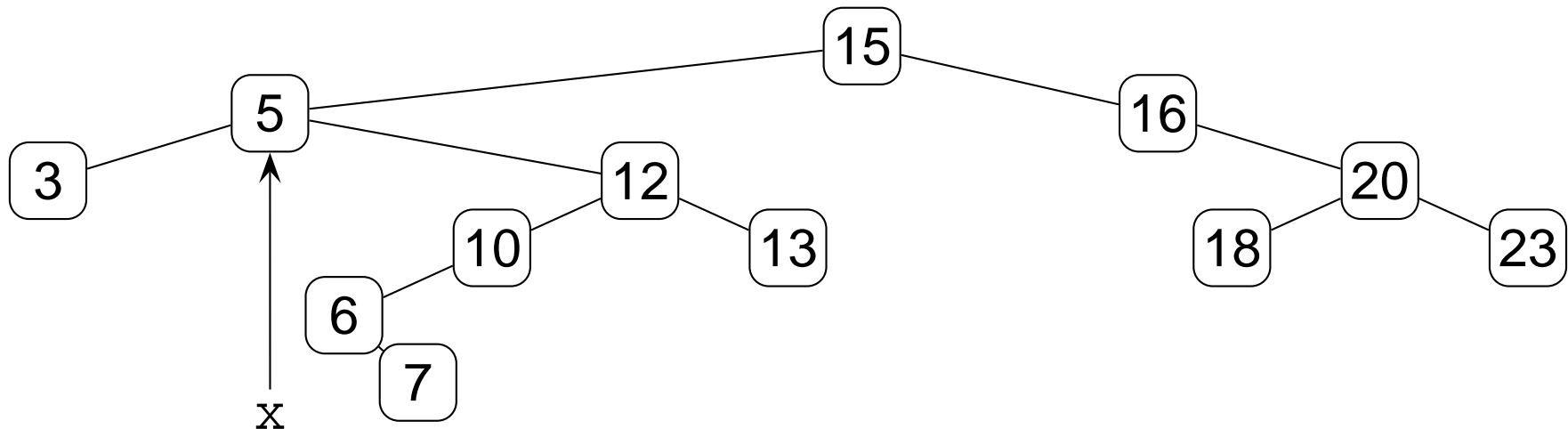
scc(Node x)



Tree Delete [target has two children] (1) | Tree Successor

scc(Node x):

```
if right(x) ≠ NIL then return treeMinimum(right(x))  
loop from y ← p(x) while y ≠ NIL, x = right(y)  
    x ← y, y ← p(y)  
return y
```



Tree Delete [target has two children] (1) | T S (1) | Tree Minimum

scc(Node x):

if right(x) ≠ NIL **then return** treeMinimum(right(x))

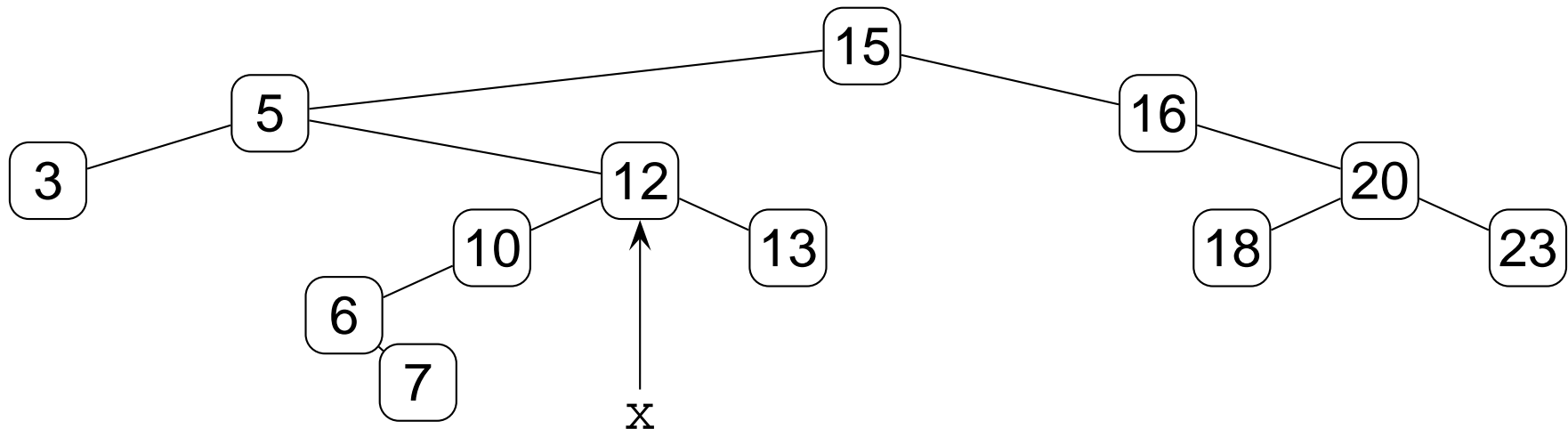
...

treeMinimum(Node x)

while left(x) ≠ NIL

 x ← left(x)

return x



Tree Delete [target has two children] (1) | T S (1) | Tree Minimum (1)

scc(Node x):

if right(x) \neq NIL **then return** treeMinimum(right(x))

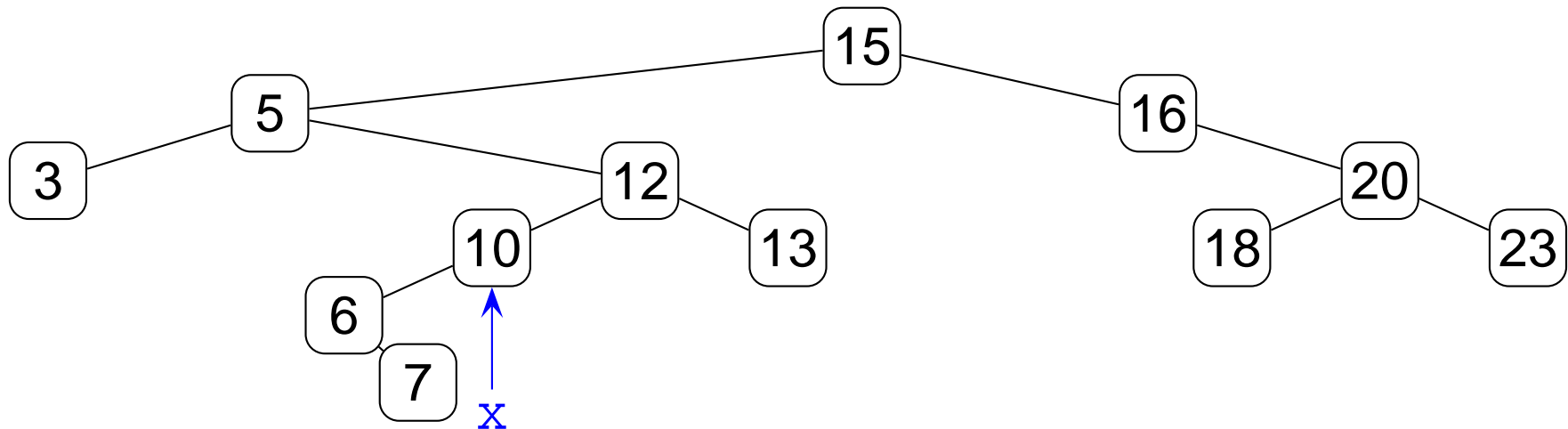
...

treeMinimum(Node x)

while left(x) \neq NIL

 x \leftarrow left(x)

return x



Tree Delete [target has two children] (1) | T S (1) | Tree Minimum (2)

scc(Node x):

```
if right(x) ≠ NIL then return treeMinimum(right(x))
```

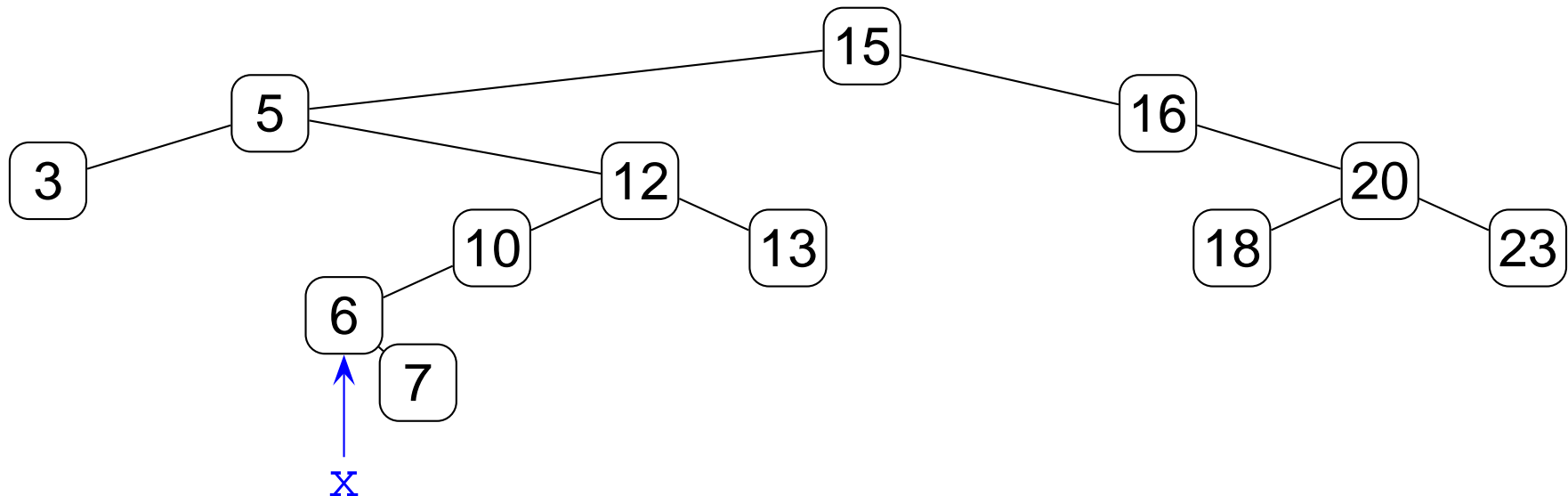
...

treeMinimum(Node x)

```
while left(x) ≠ NIL
```

```
  x ← left(x)
```

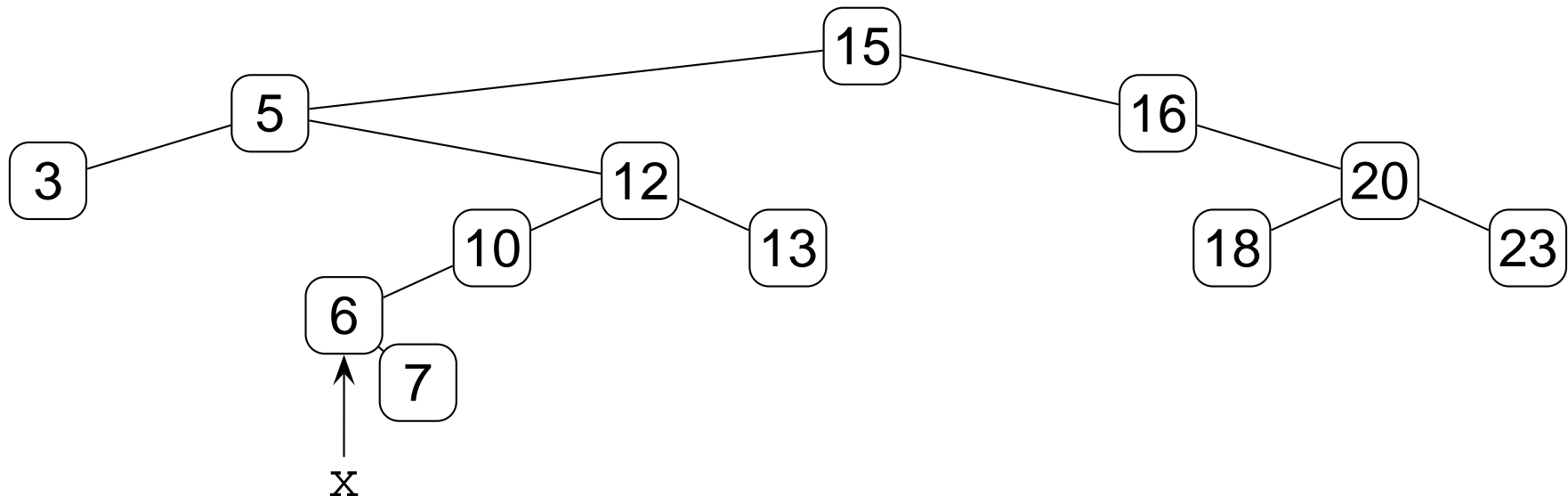
```
return x
```



Tree Delete [target has two children] (1) | Tree Successor (2)

scc(Node x):

```
if right(x)  $\neq$  NIL then return treeMinimum(right(x))  
loop from y  $\leftarrow$  p(x) while y  $\neq$  NIL, x = right(y)  
    x  $\leftarrow$  y, y  $\leftarrow$  p(y)  
return y
```



Tree Delete [target has two children] (1)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

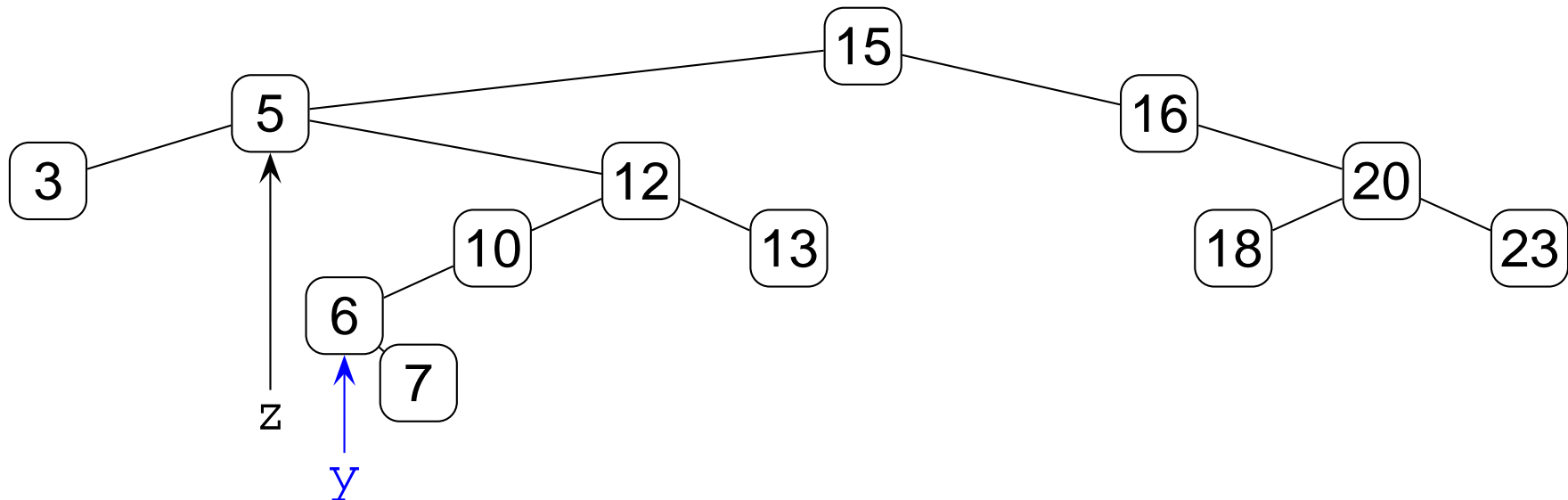
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has two children] (2)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

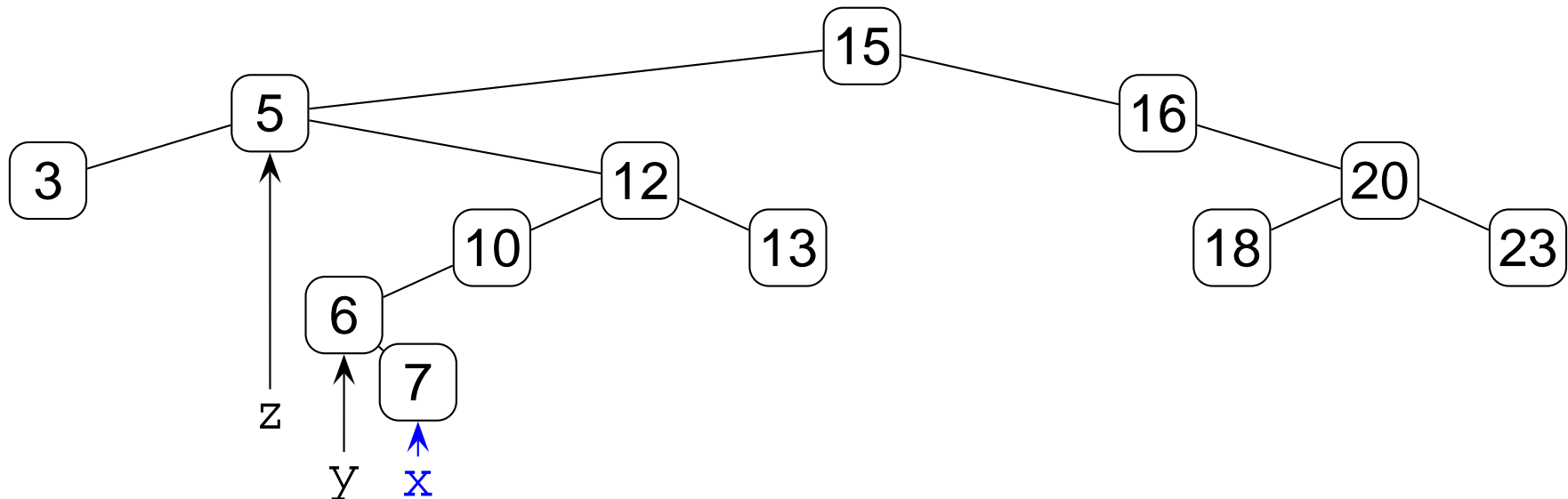
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has two children] (3)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

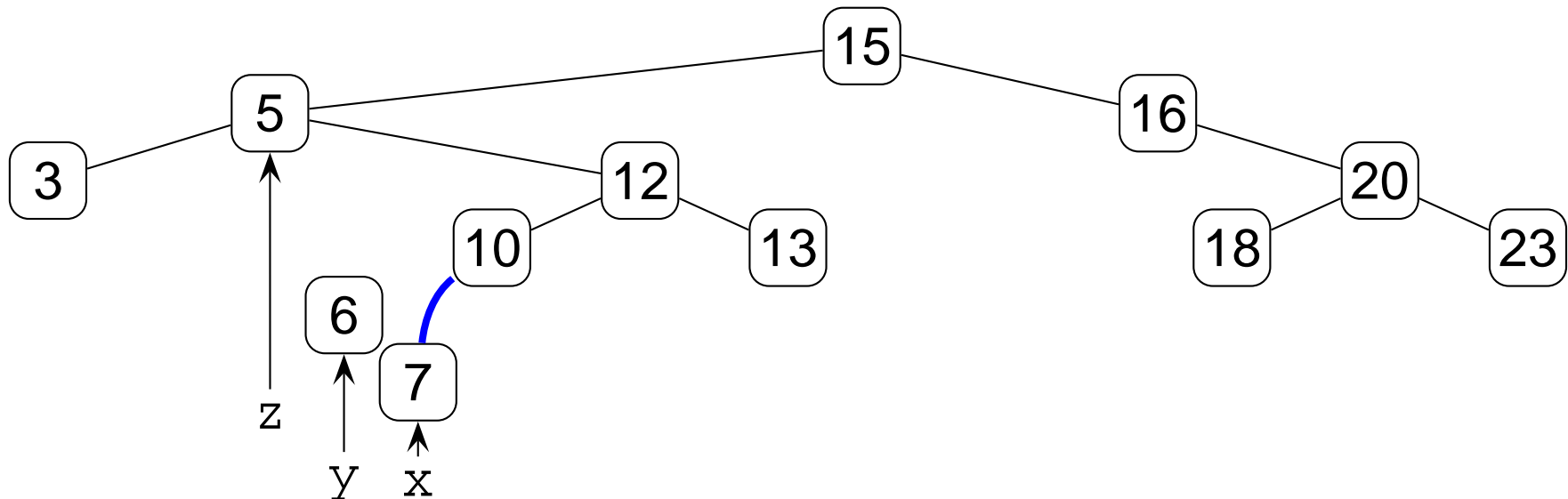
if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$



Tree Delete [target has two children] (done)

treeDelete(Tree T, Node z):

if left(z)=NIL or right(z)=NIL **then** $y \leftarrow z$ **else** $y \leftarrow \text{scc}(z)$

if left(y)≠NIL **then** $x \leftarrow \text{left}(y)$ **else** $x \leftarrow \text{right}(y)$

if $x \neq \text{NIL}$ **then** $p(x) \leftarrow p(y)$

if $p(y) = \text{NIL}$ **then** $\text{root}(T) \leftarrow x$

else if $y = \text{left}(p(y))$ **then** $\text{left}(p(y)) \leftarrow x$

else $\text{right}(p(y)) \leftarrow x$

if $y \neq z$ **then** $\text{key}(z) \leftarrow \text{key}(y)$, $\text{data}(z) \leftarrow \text{data}(y)$

