

## BST Traversal

Inorder traversal:

Tree searching

$x$  is a node  
key

Tree-search ( $x, K$ )

if  $K = x.\text{key}$  or  $x == \text{Null}$   
return  $x$

if  $K < x.\text{key}$

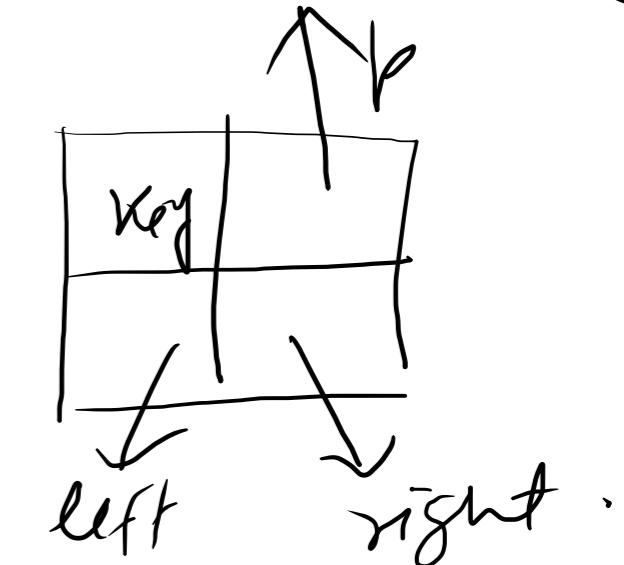
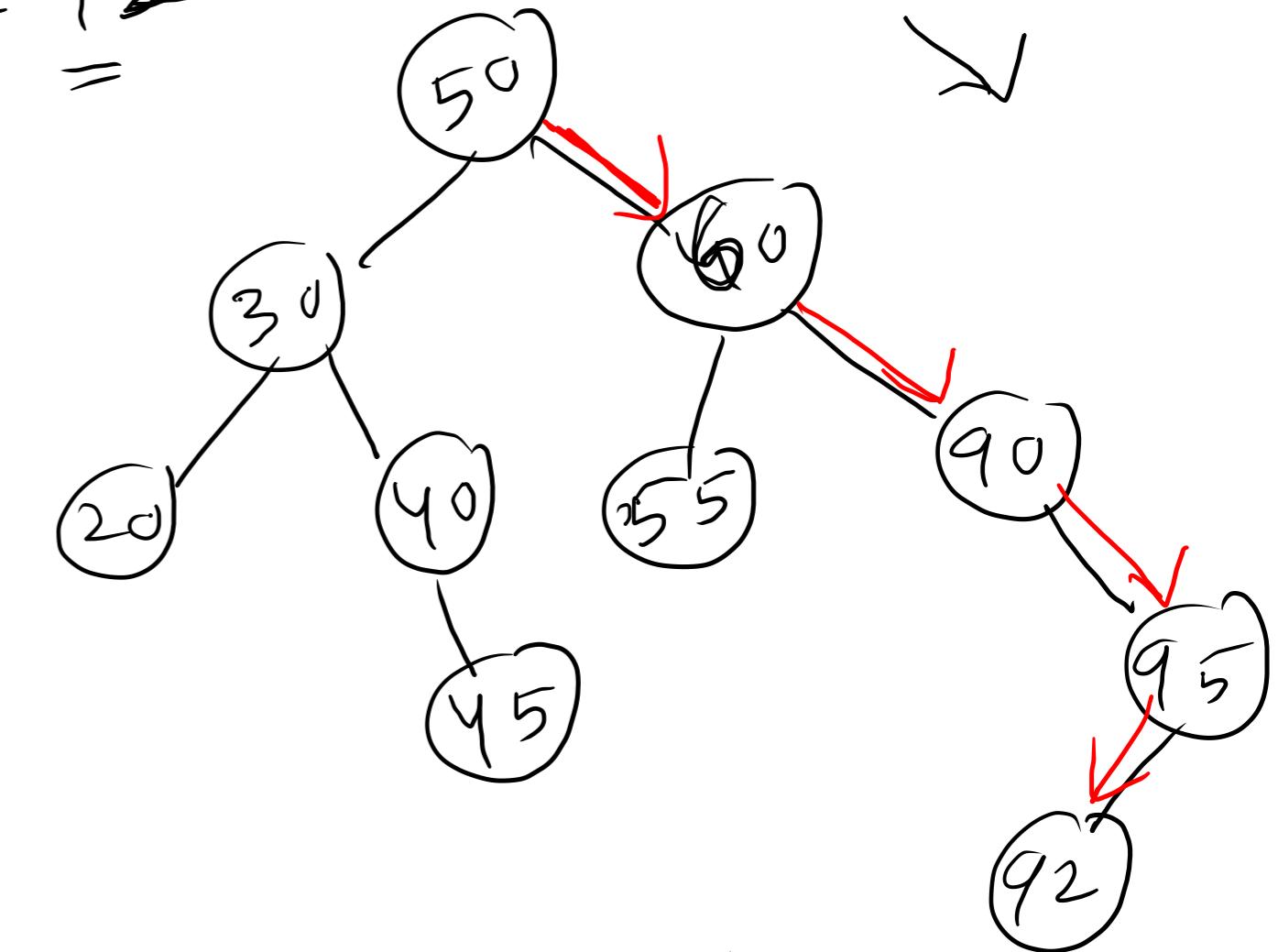
tree-search ( $x.\text{left}, K$ )

else

tree-search ( $x.\text{right}, K$ )

Running time:  $O(\text{height})$

$$K = 92$$



## Iterative

iter-tree-search( $x, k$ )

while  $x \neq \text{Null}$  or  $x.\text{key} \neq k$

    if  $k < x.\text{key}$

$x = x.\text{left}$

    else

$x = x.\text{right}$

return  $x$

Running time:  $O(\text{height})$

## Not-exact search

not-exact-search( $x, k$ )

$y = \text{null}$

while  $x \neq \text{null}$

$y = x$

if  $k < x.\text{key}$

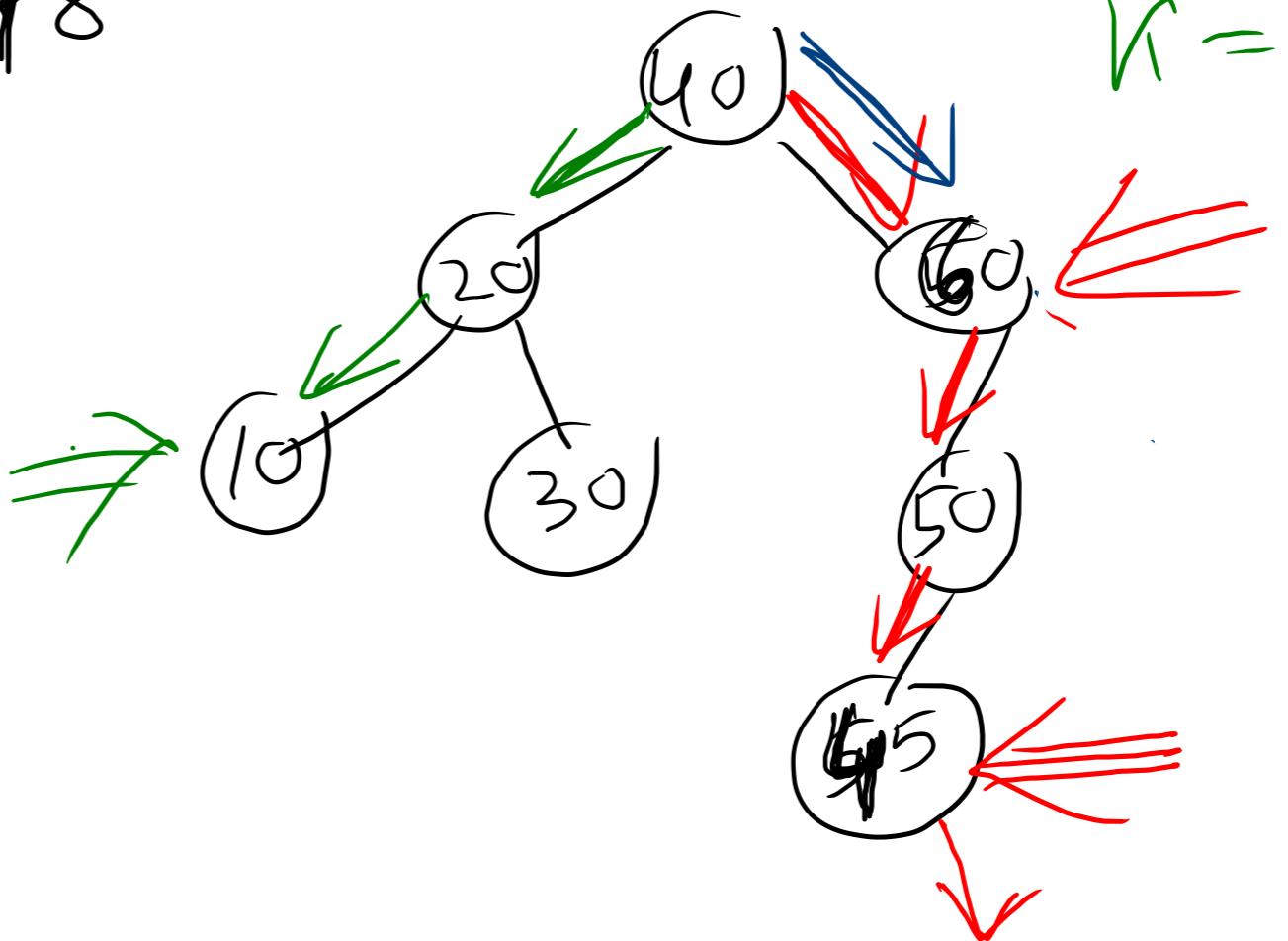
$x = x.\text{left}$

else

$x = x.\text{right}$

return  $y$

$K = 48$



$K = 18$

Running time:  $O(\text{height})$

# Tree minimum and maximum

Tree minimum ( $x$ )

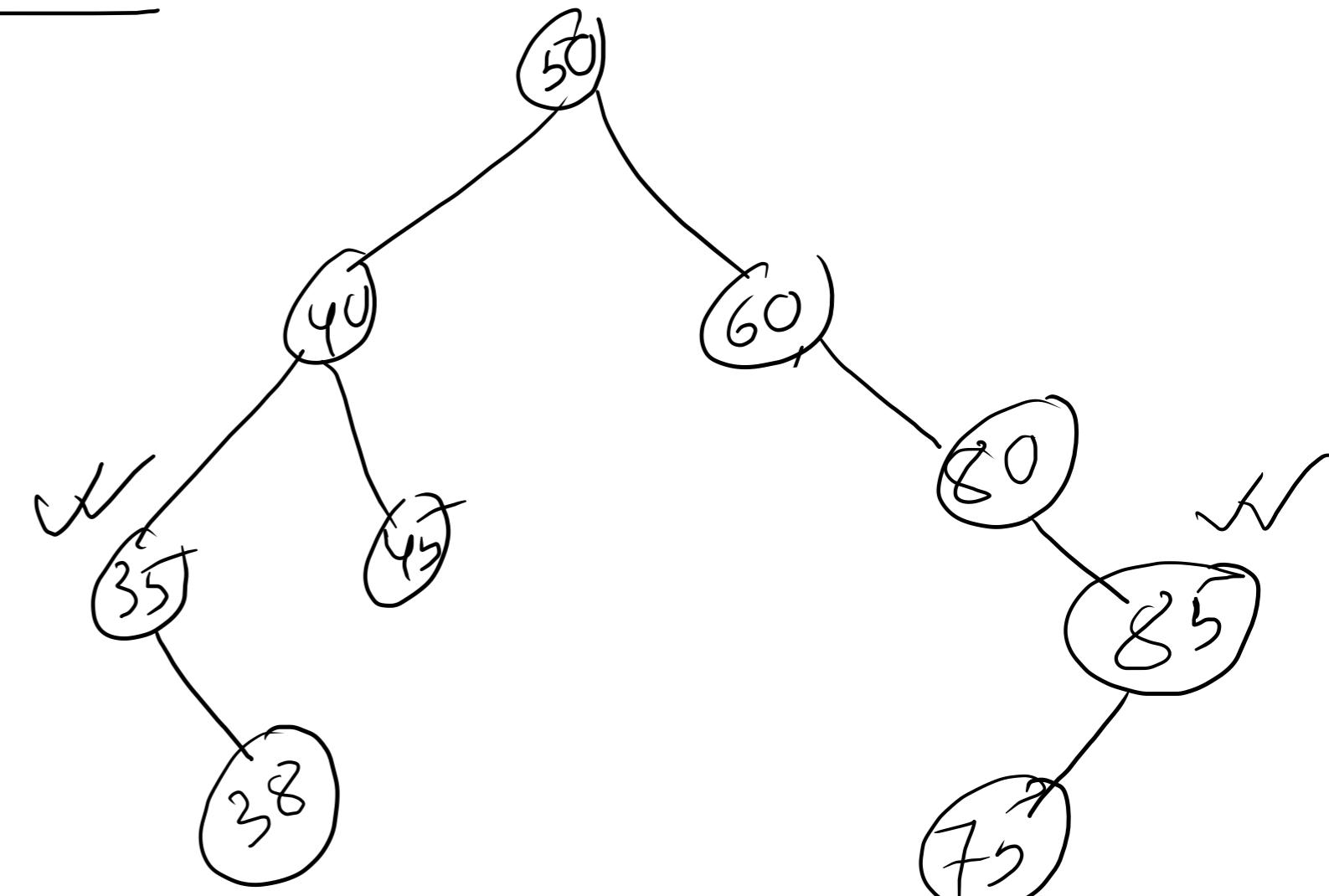
```
while  $x \cdot \text{left} \neq \text{Null}$ 
     $x = x \cdot \text{left}$ 
```

return  $x$

Tree maximum ( $x$ )

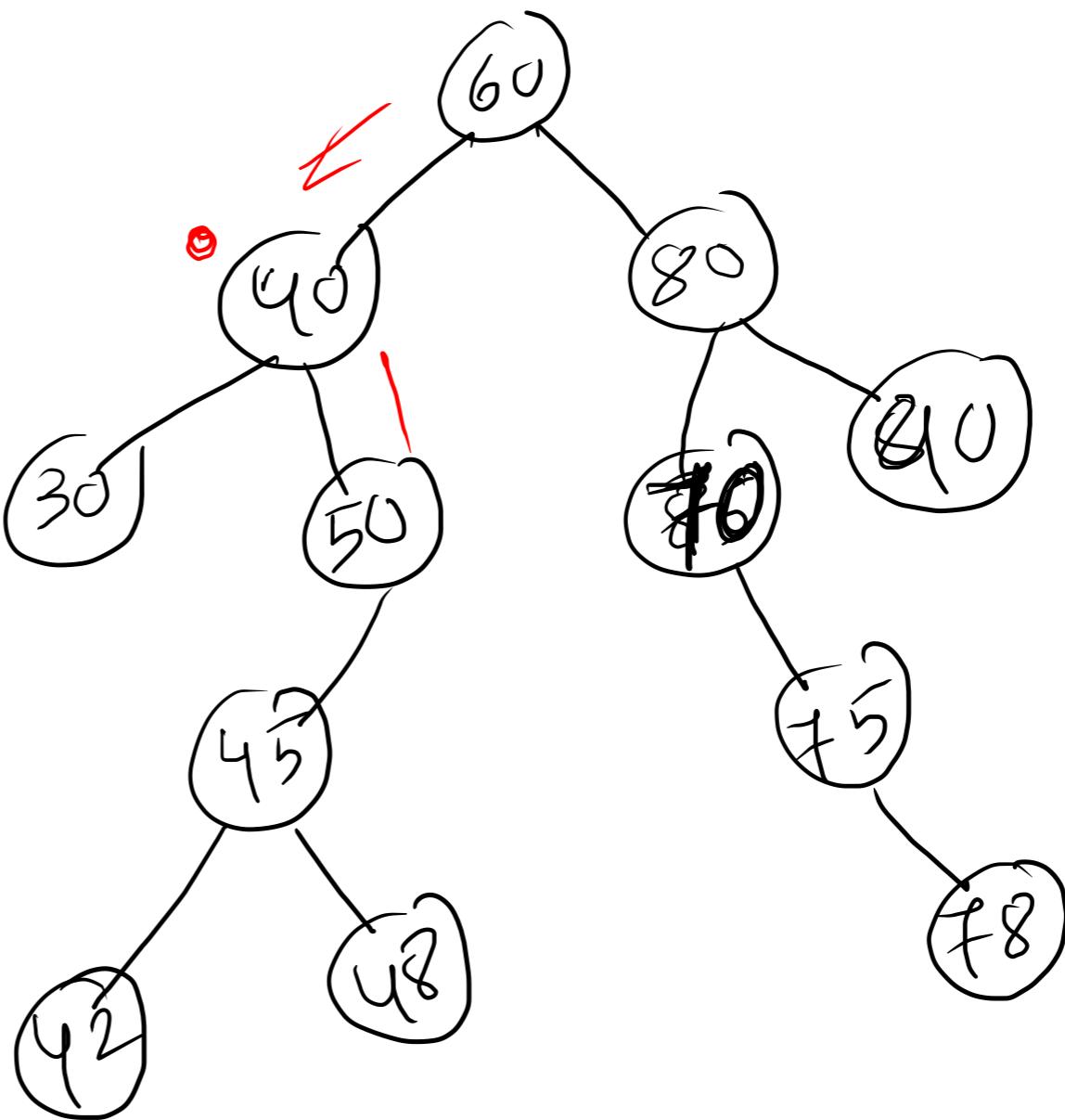
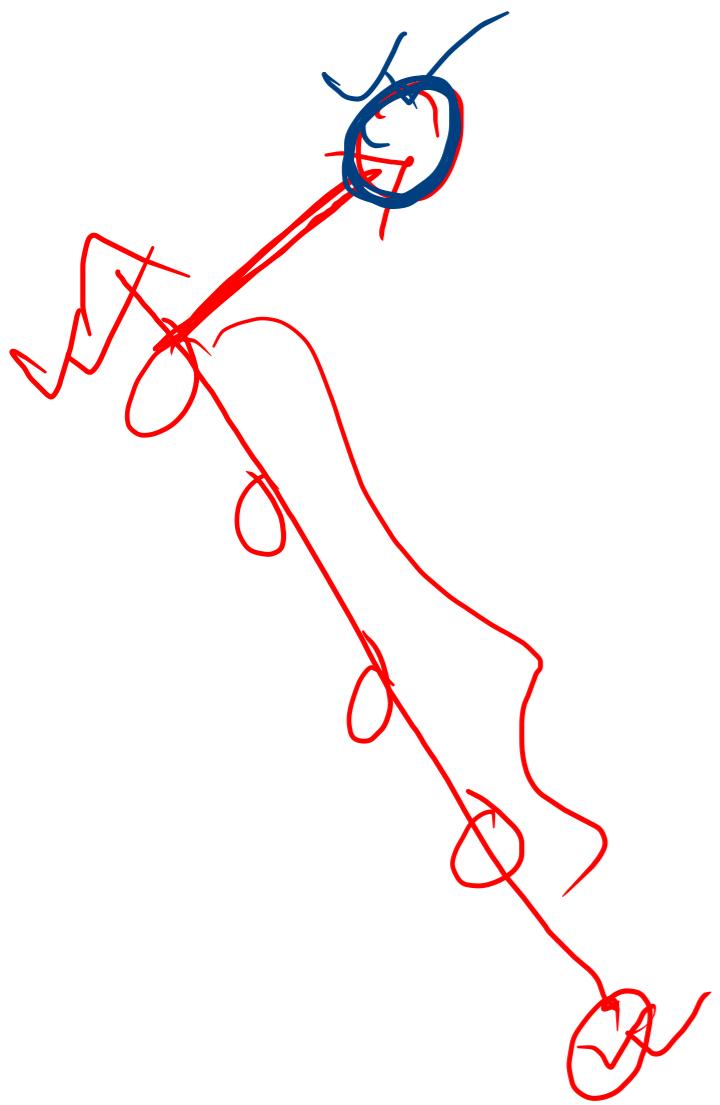
```
while  $x \cdot \text{right} \neq \text{Null}$ 
     $x = x \cdot \text{right}$ 
```

return  $x$



time:  $O(\text{height})$

## Successor



45 — 48  
40 — 42  
450 — 60  
f8 — 80  
90 — null

Tree-successor( $x$ )

if  $x.\text{right} \neq \text{null}$

    return tree-minimum( $x.\text{right}$ )

$y = x.p$

while  $y \neq \text{null}$  and  $x \neq y.\text{left}$

$x = y$

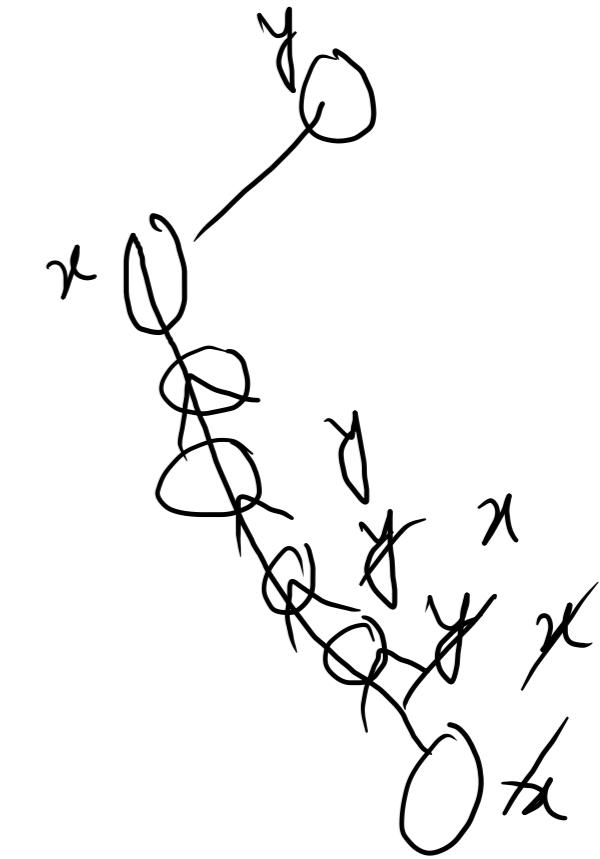
$y = x.p$

return  $y$

Tree-predecessor

Running time:  $O(\text{height})$

~~HW.~~



## Dynamic operations

Insert - 78  
Insert - 74

### BST tree insertion and tree deletion

Tree insertion  
tree  $\xrightarrow{\text{tree } T}$  node.

Tree-insertion ( $T, z$ )

$x = T, \text{root}$

$y = \text{null}$

while  $x \neq \text{null}$

$y = x$

if  $z.\text{key} < x.\text{key}$

$x = x.\text{left}$

else

$x = x.\text{right}$

$\Rightarrow$  if  $y == \text{null}$   
 $T.\text{root} = z$

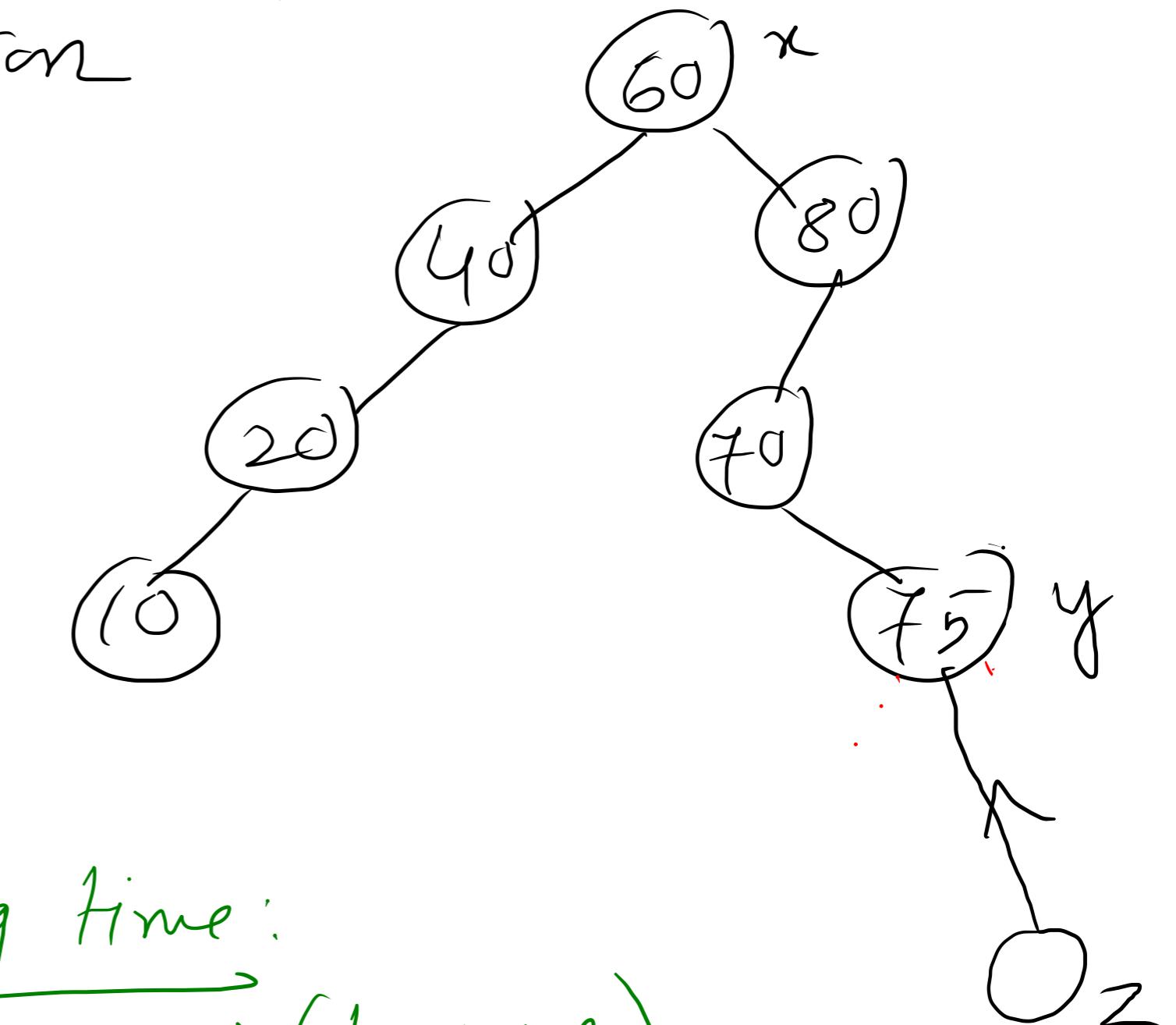
$z.p = y$

if  $z.\text{key} < y.\text{key}$

$y.\text{left} = z$

else

$y.\text{right} = z$



Running Time:

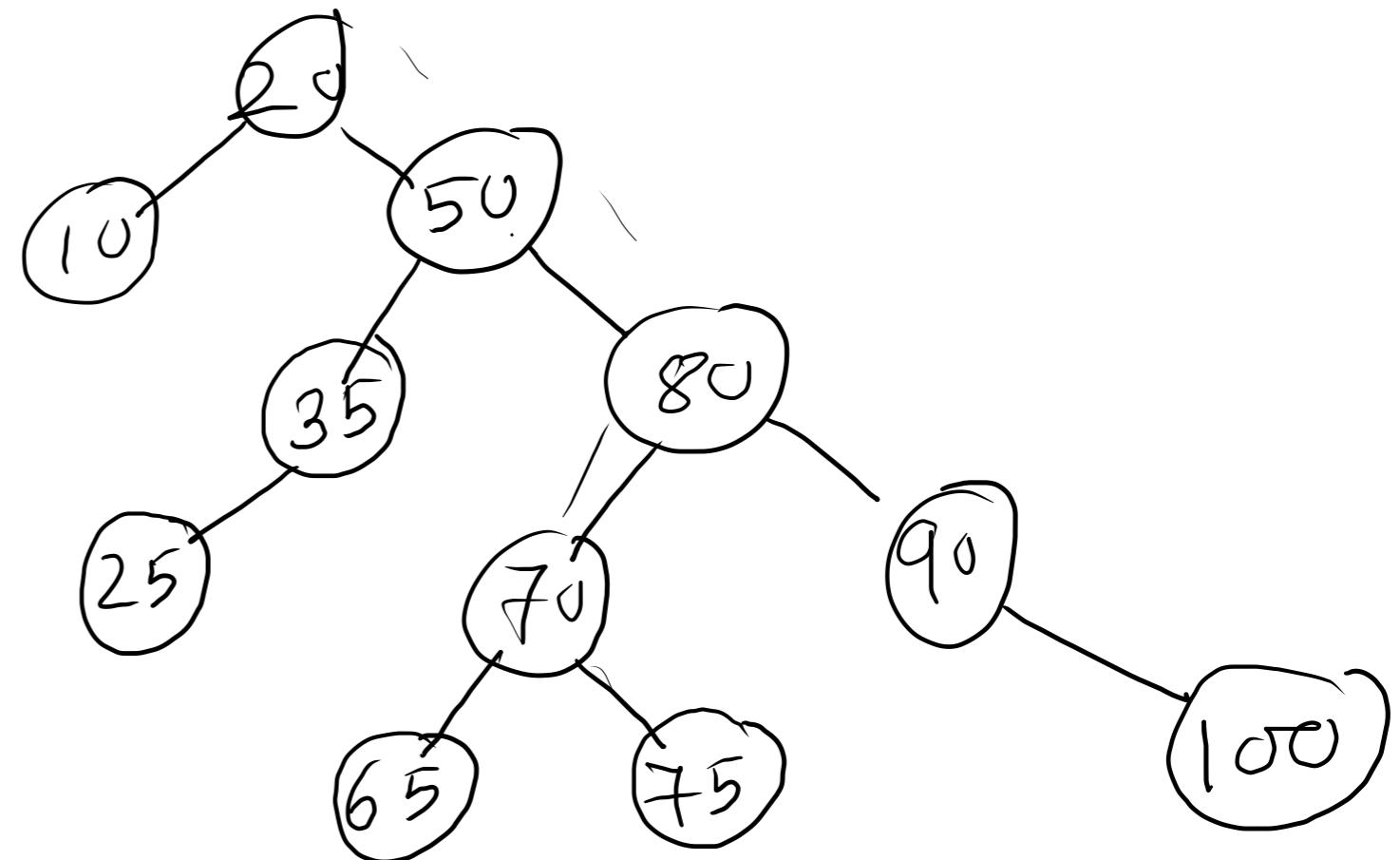
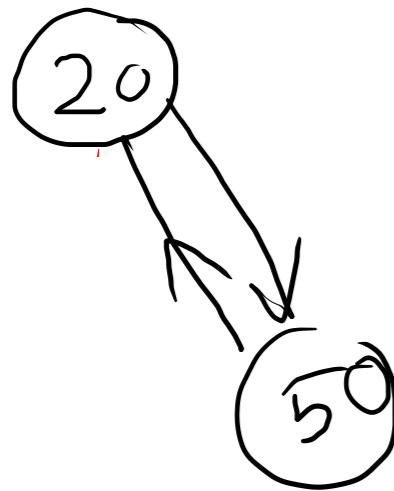
$O(\text{height})$

$\log n \leq \text{height} \leq n$

insert

20, 50, 80, 70, 90, 10, 65, 75  
35, 25, 100

root



5

20

10

35

6

2

9

6

4

BST sorting running time

---

 $O(n \cdot h_{max})$