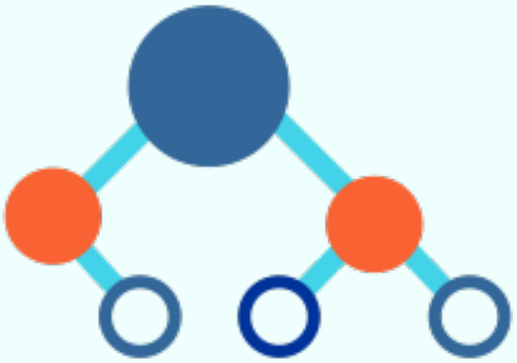
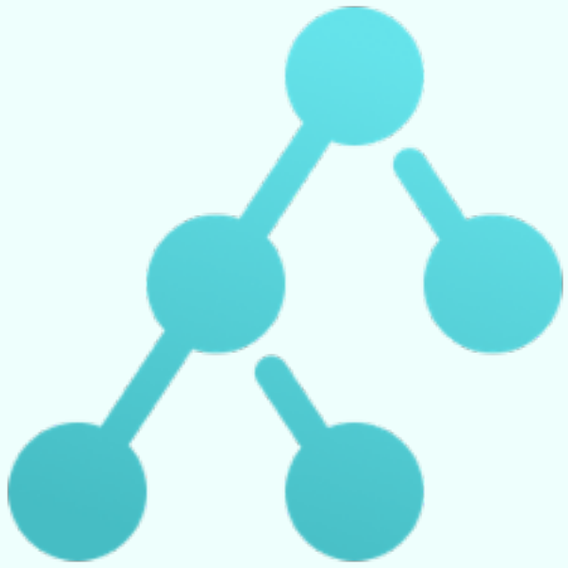
Data Structures

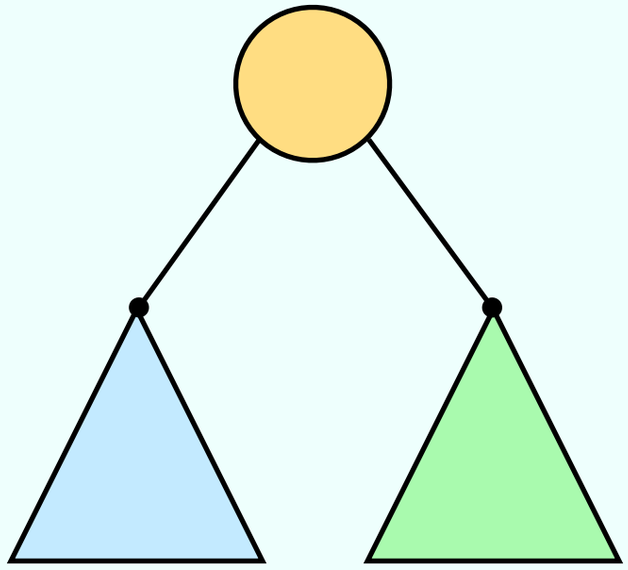
Exercise Session



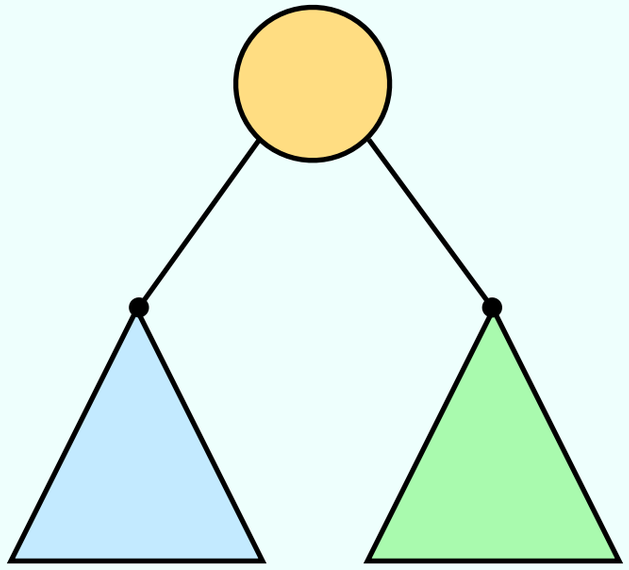
Marco Vassena

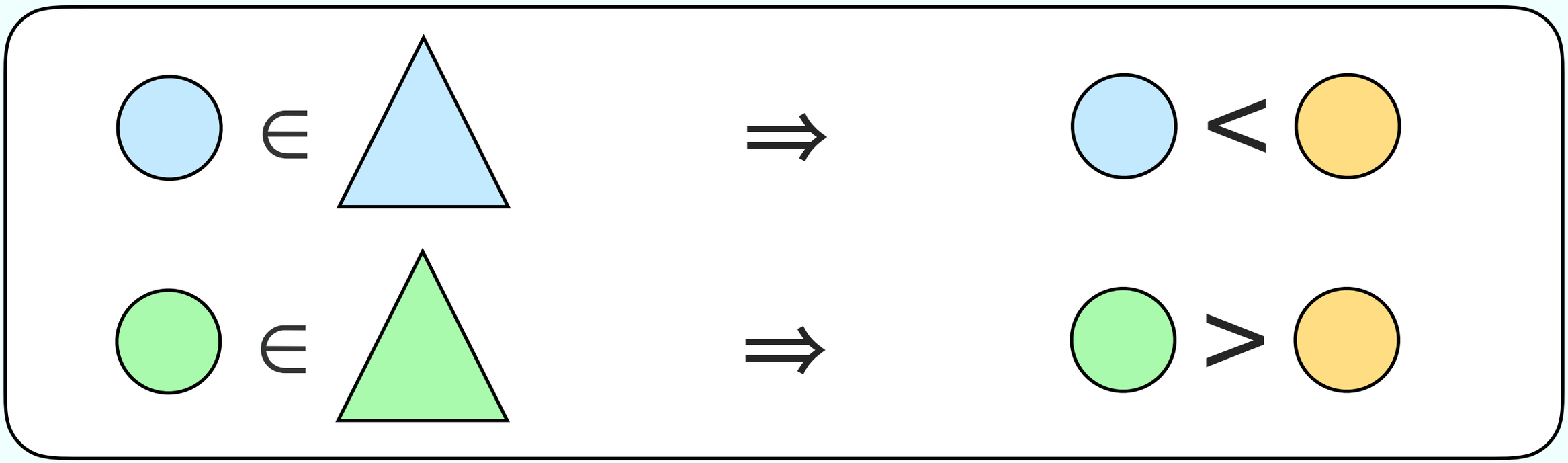


Binary Search Tree



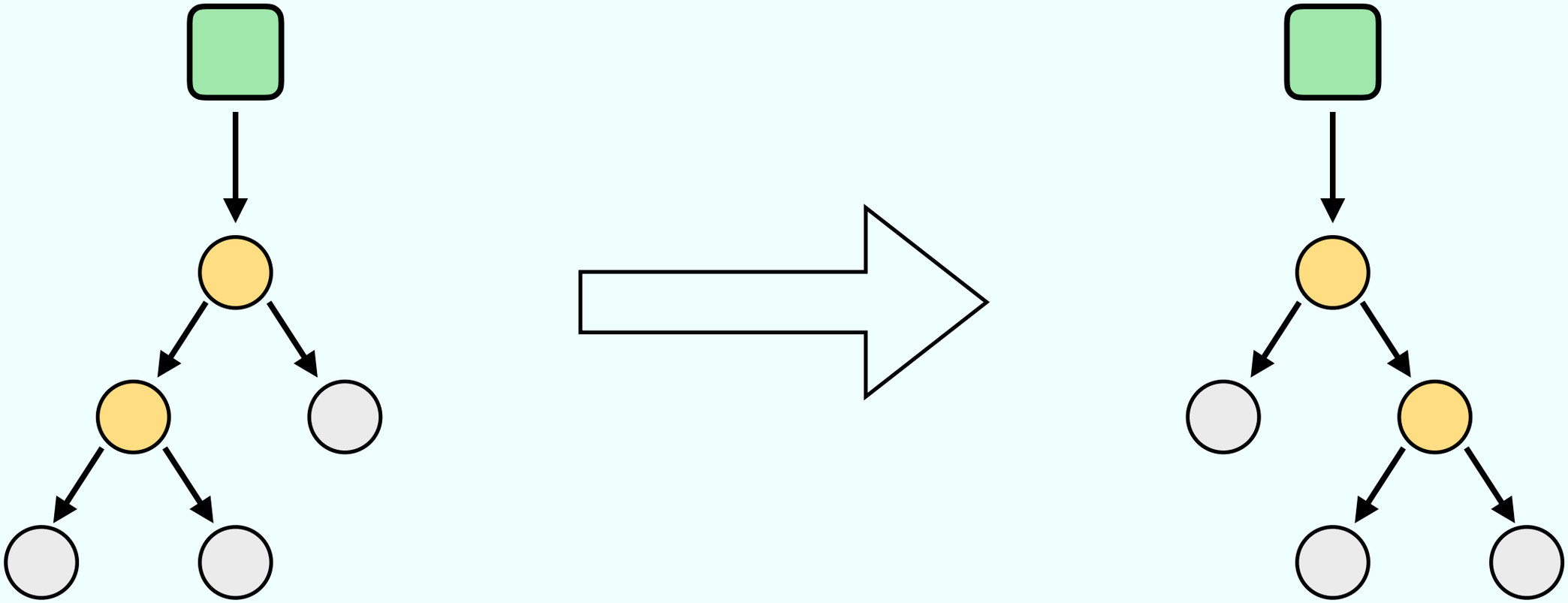
Binary Search Tree





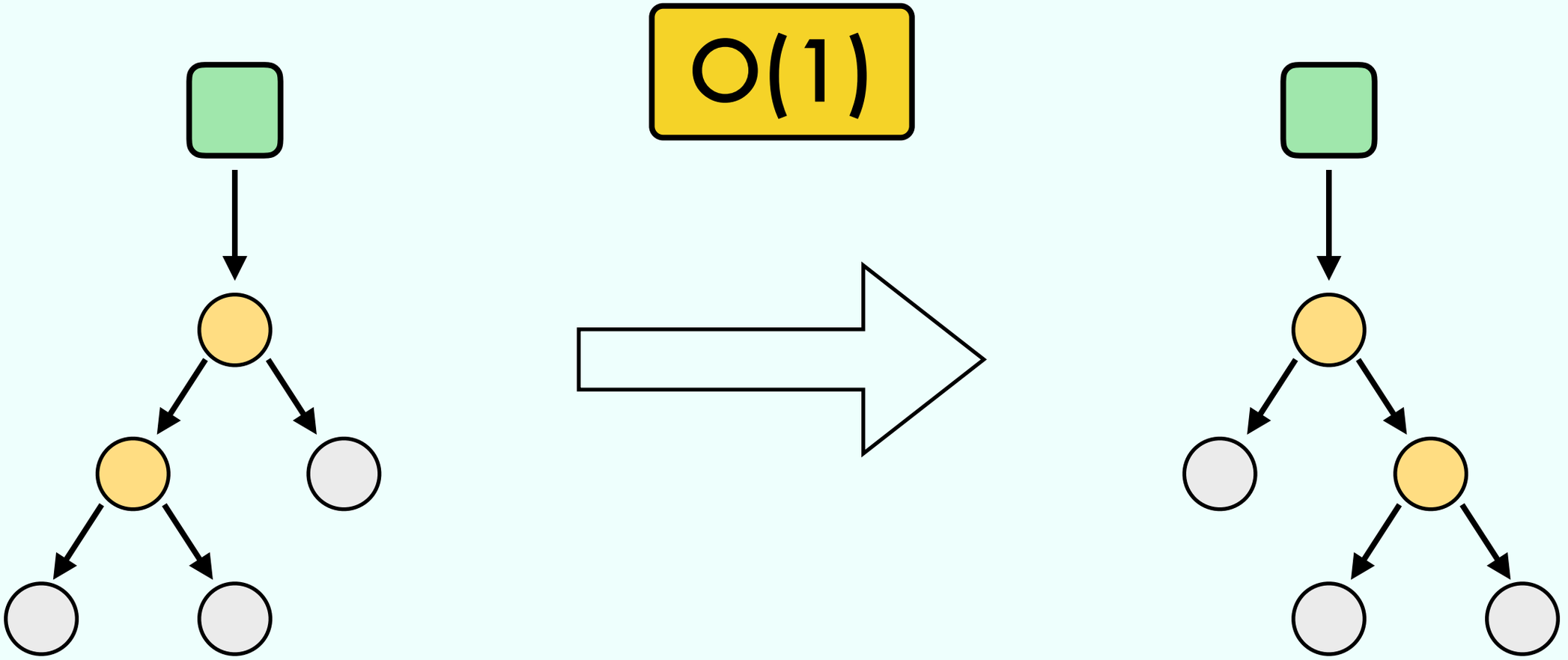
Problem 9

Reverse a search binary tree



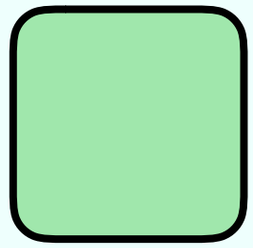
Problem 9

Reverse a search binary tree

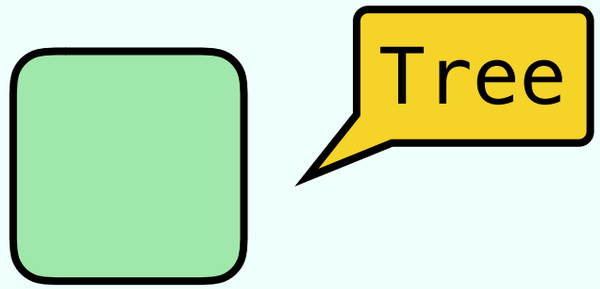


Binary Search Tree

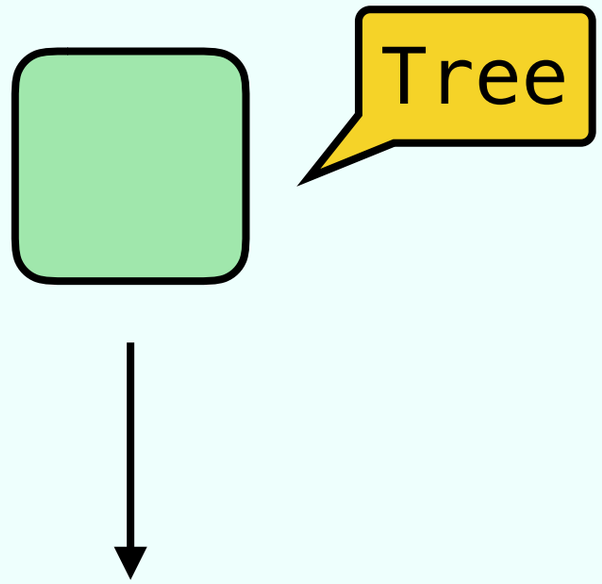
Binary Search Tree



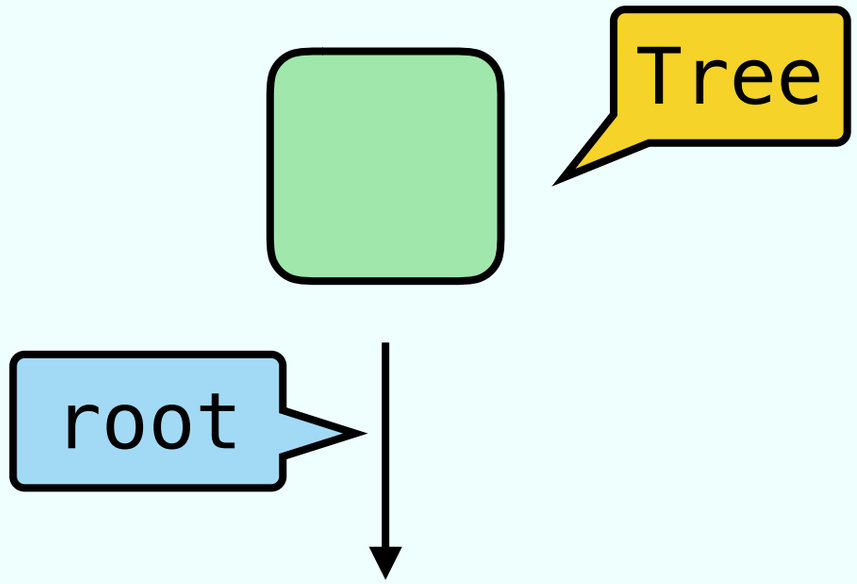
Binary Search Tree



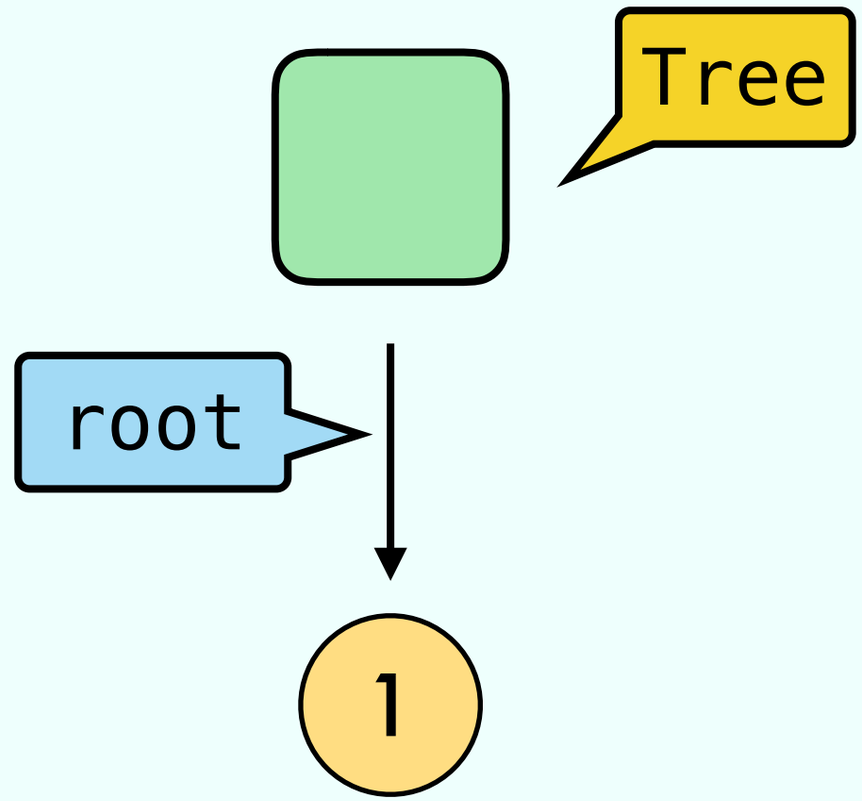
Binary Search Tree



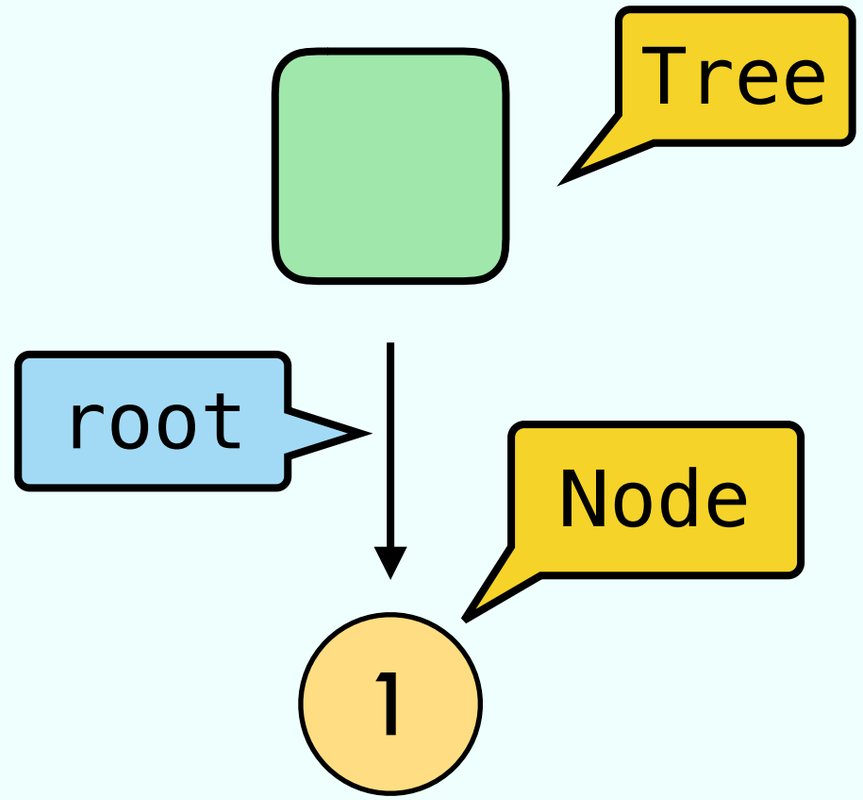
Binary Search Tree



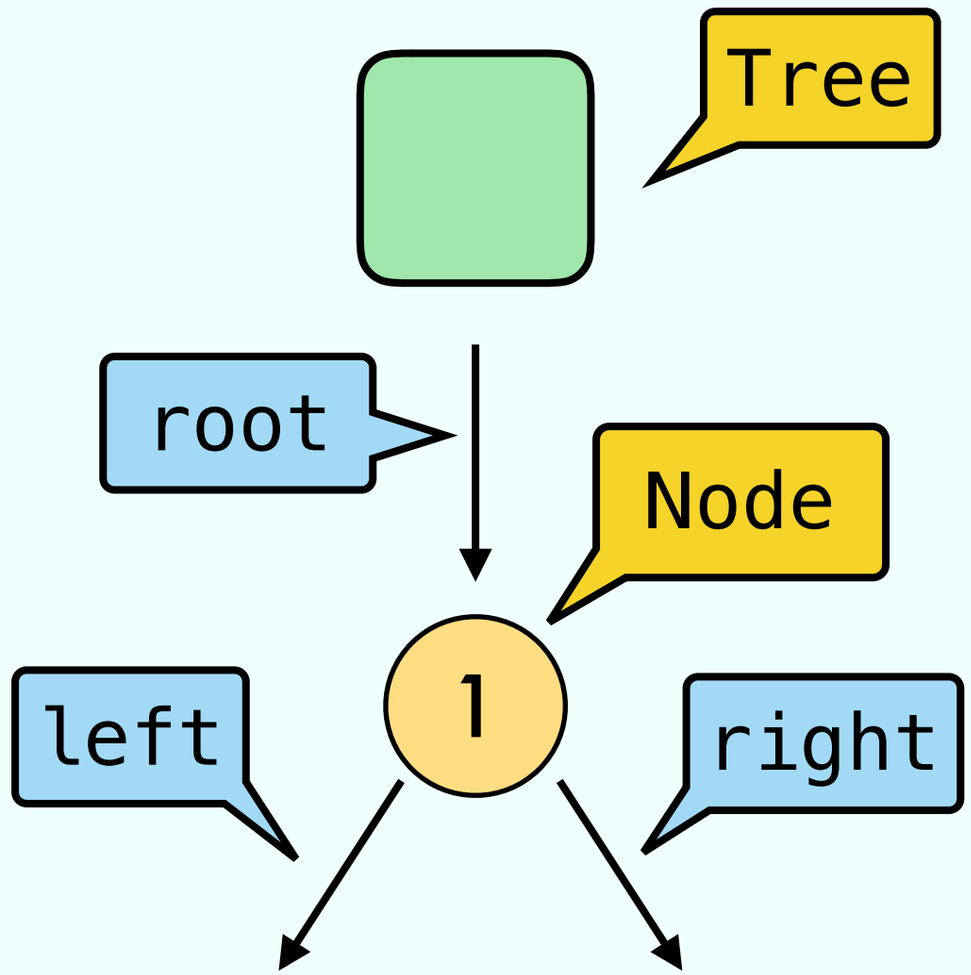
Binary Search Tree



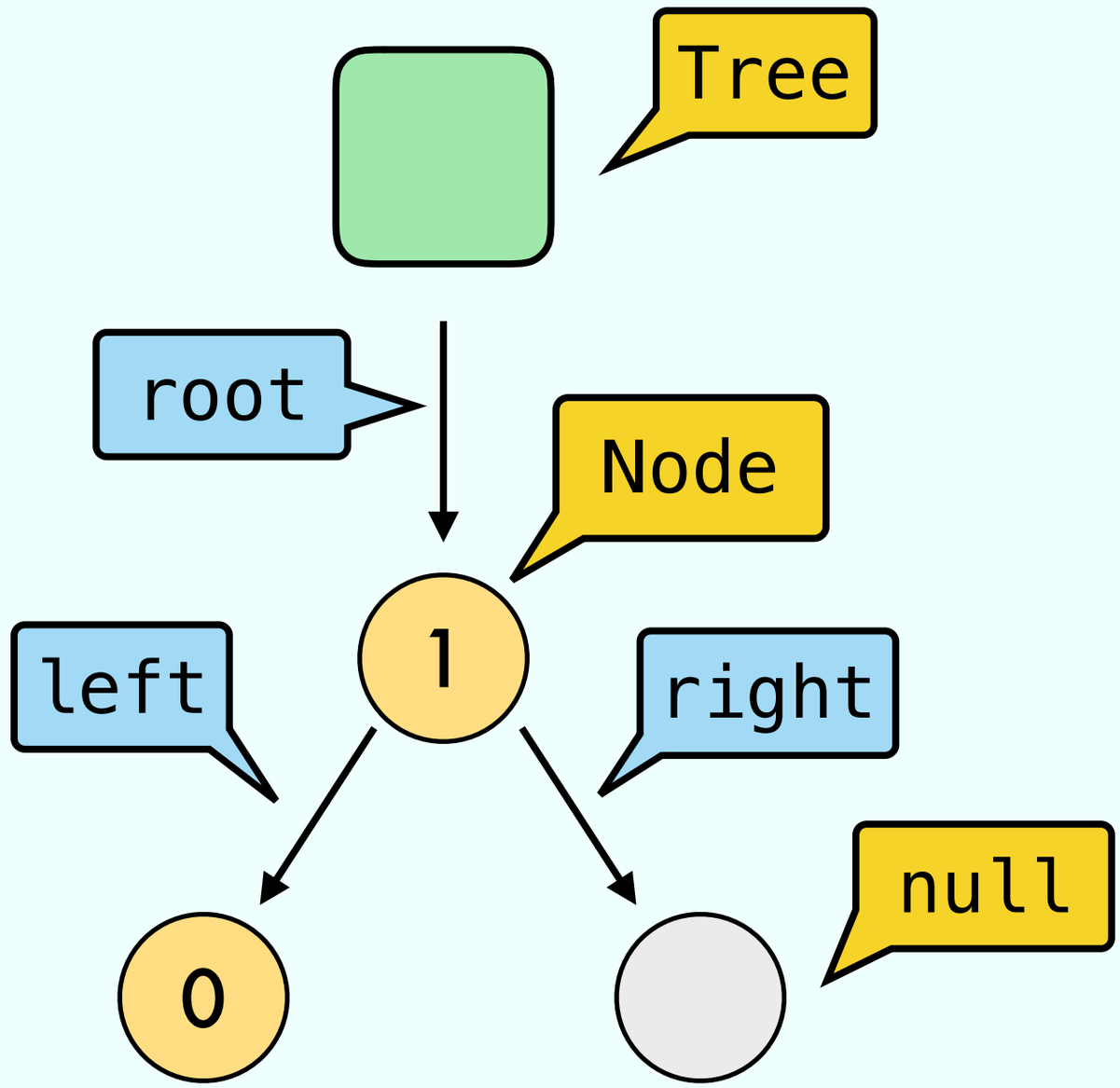
Binary Search Tree



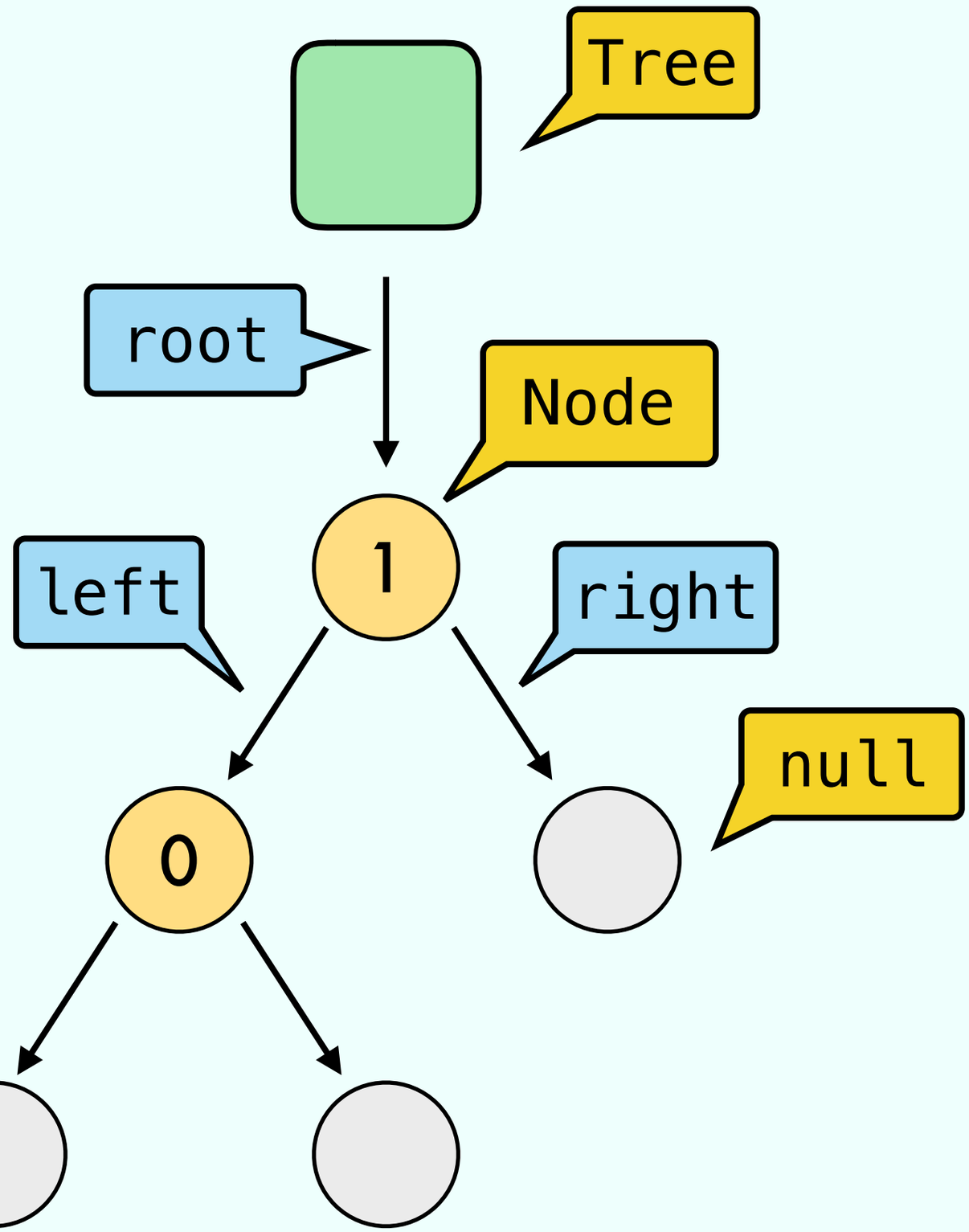
Binary Search Tree



Binary Search Tree

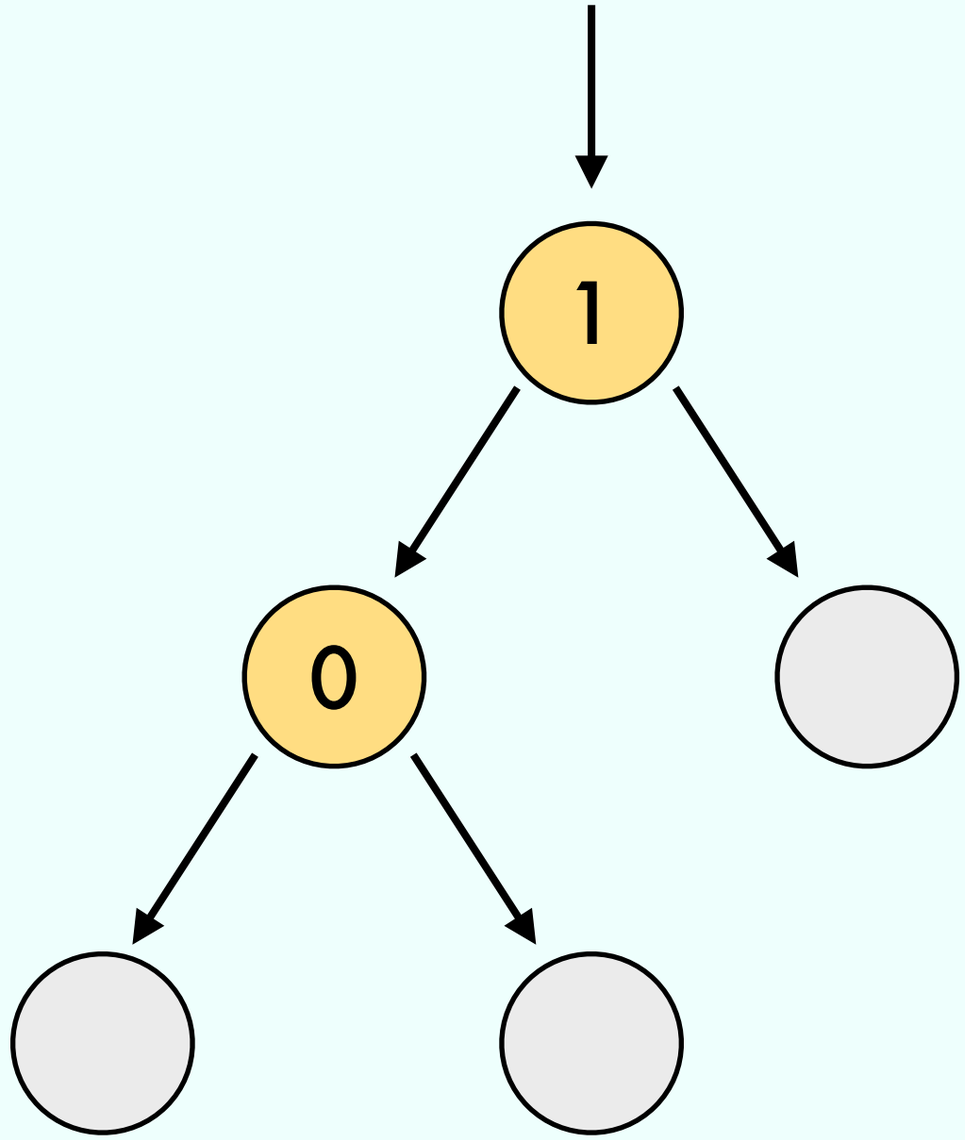


Binary Search Tree



Reversible Binary Search Tree

```
Comparator <T> cmp;
```



Analyze the time complexity

```
Tree t = new BST()
for (i = 0; i < N; i++)
    t.insert(i)
```

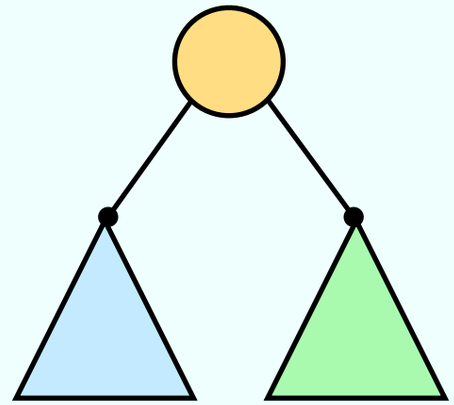
in terms of N

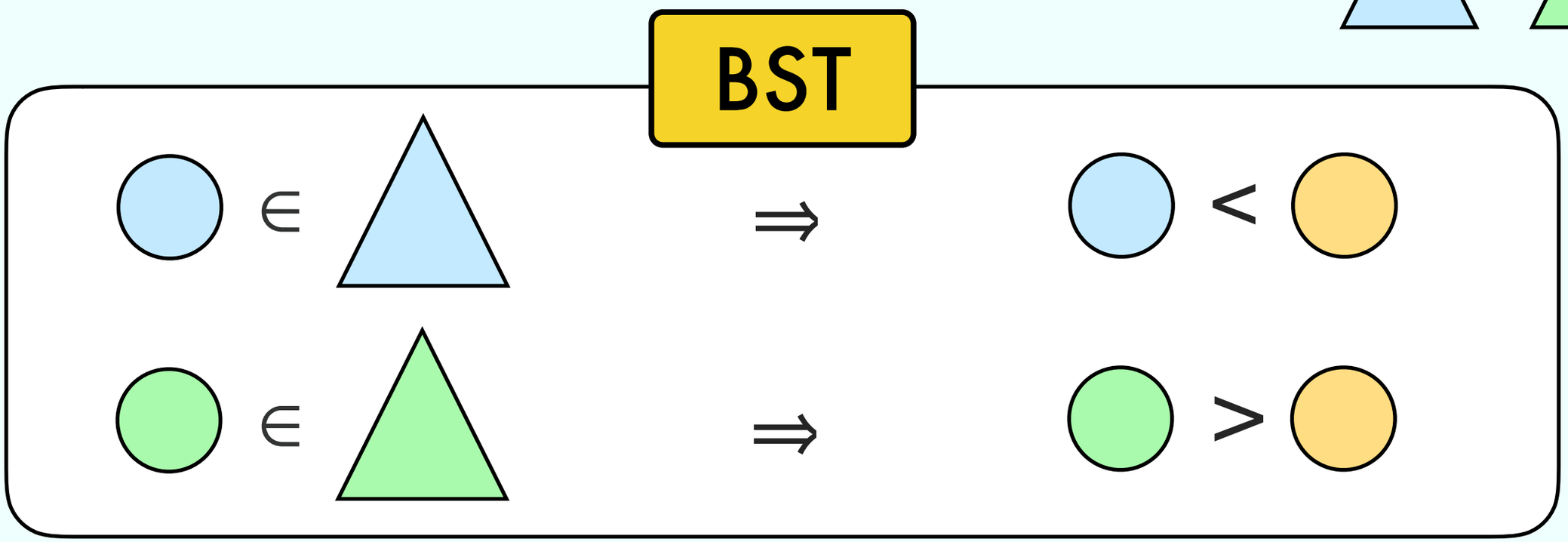
Analyze the time complexity

```
Tree t = new AVLTree()
for (i = 0; i < N; i++)
   for (j = 0; j < i; j++)
      t.insert(N)
```

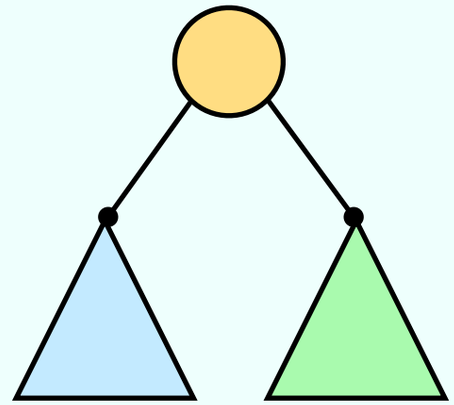
in terms of N

AVL Tree

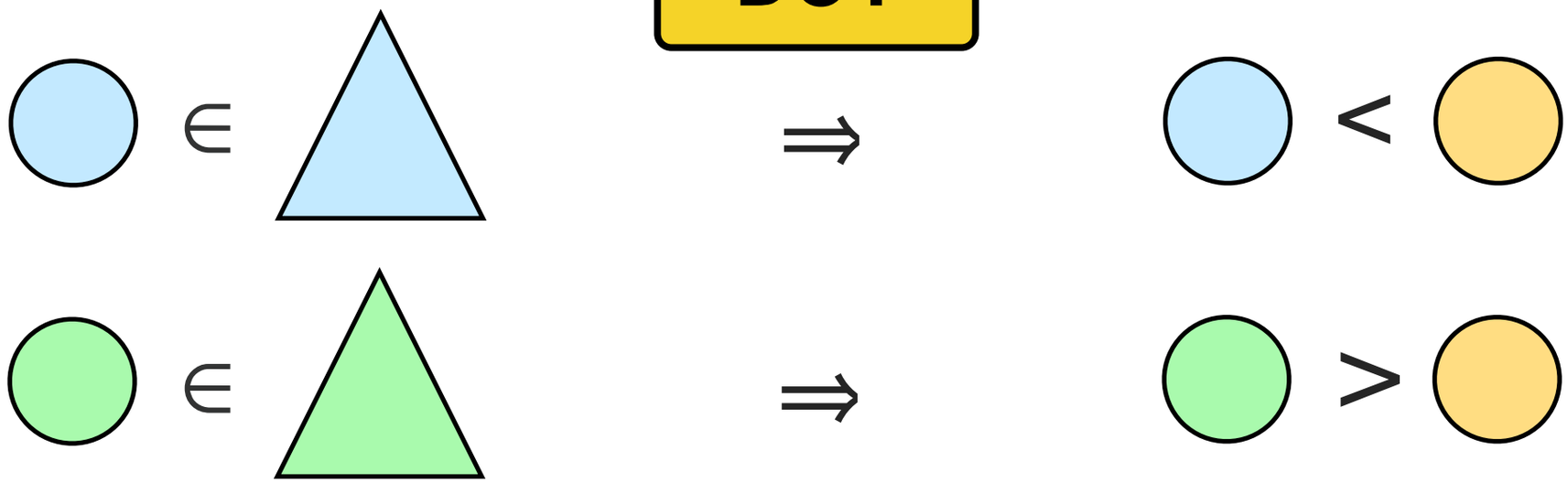




AVL Tree

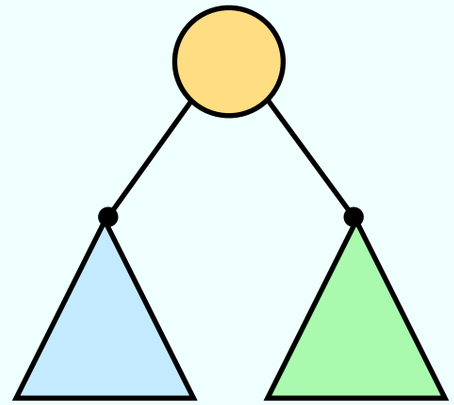


BST

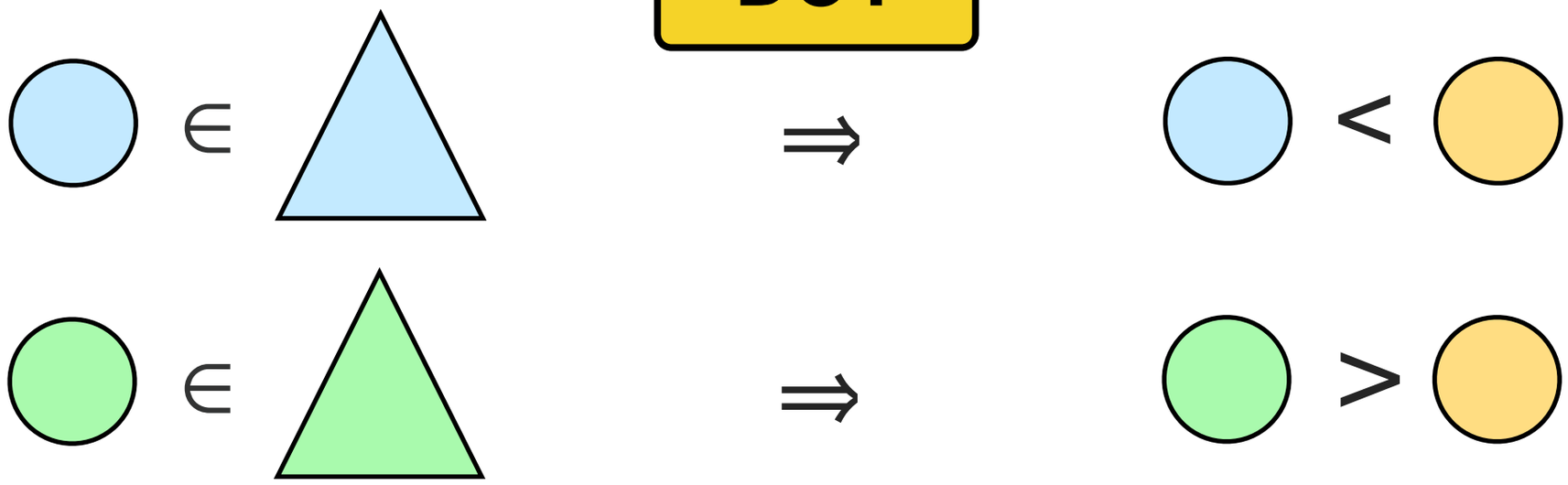


Balanced

AVL Tree



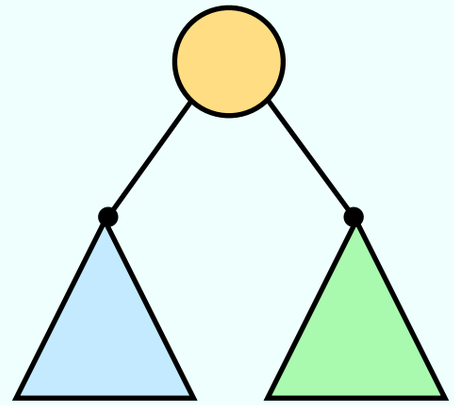
BST



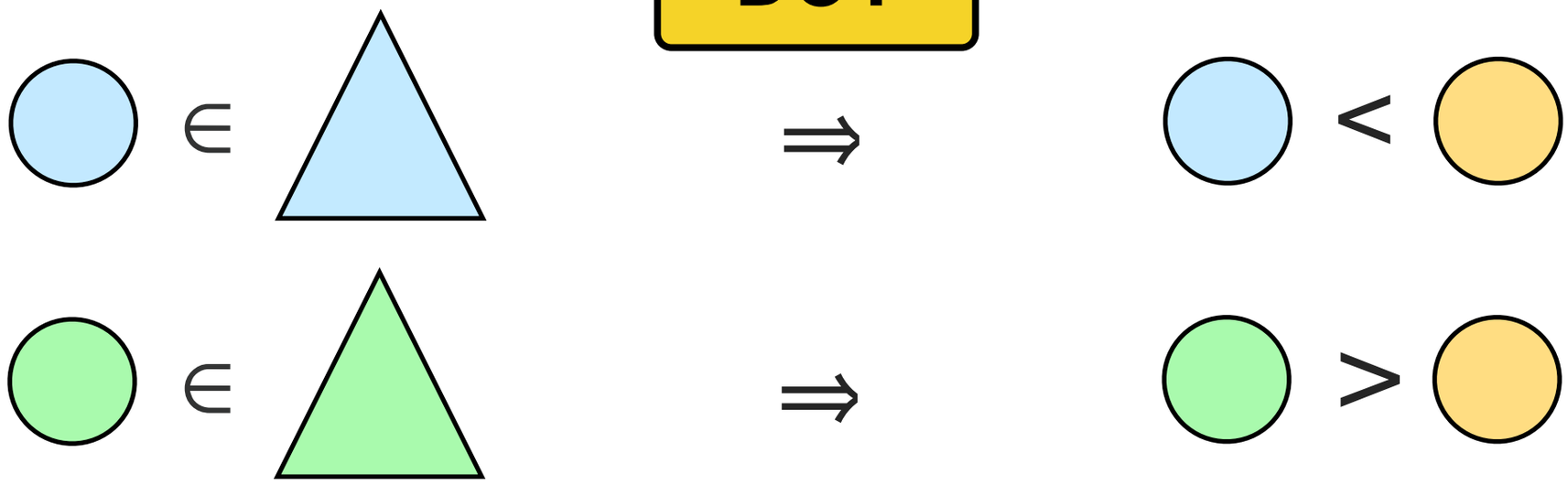
Balanced

$$|h(\triangle) - h(\triangle)| \leq 1$$

AVL Tree



BST

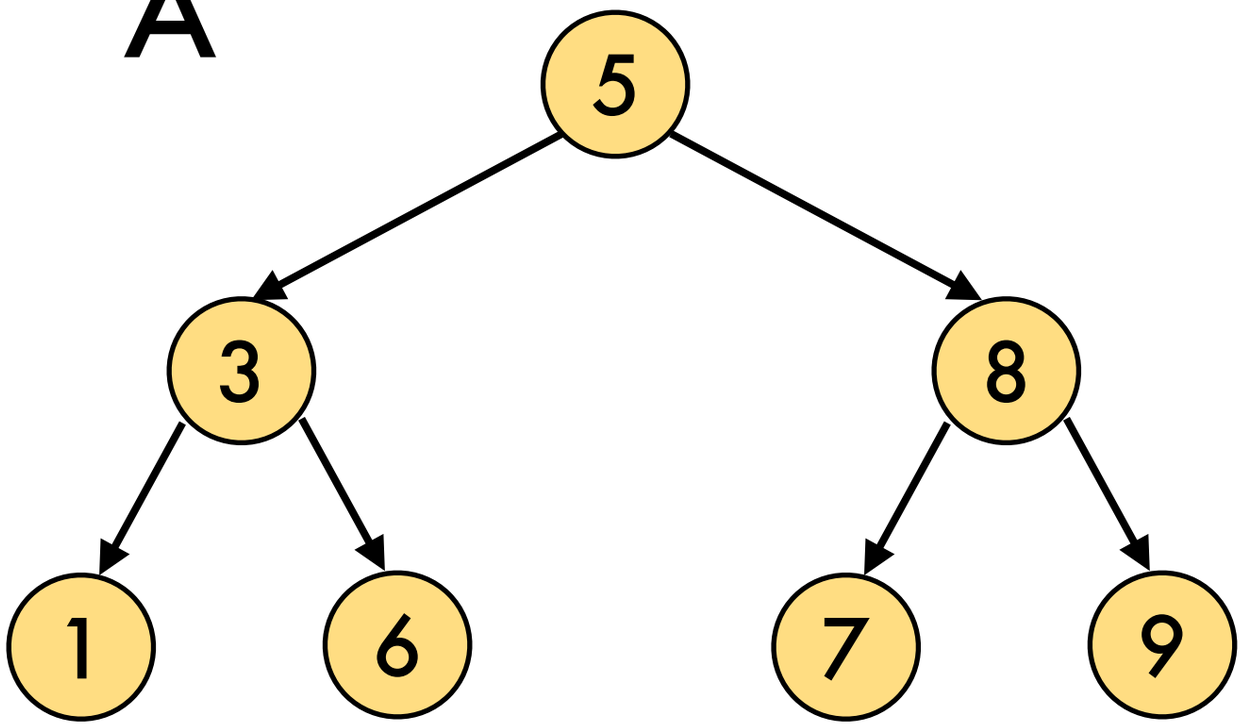


Balanced

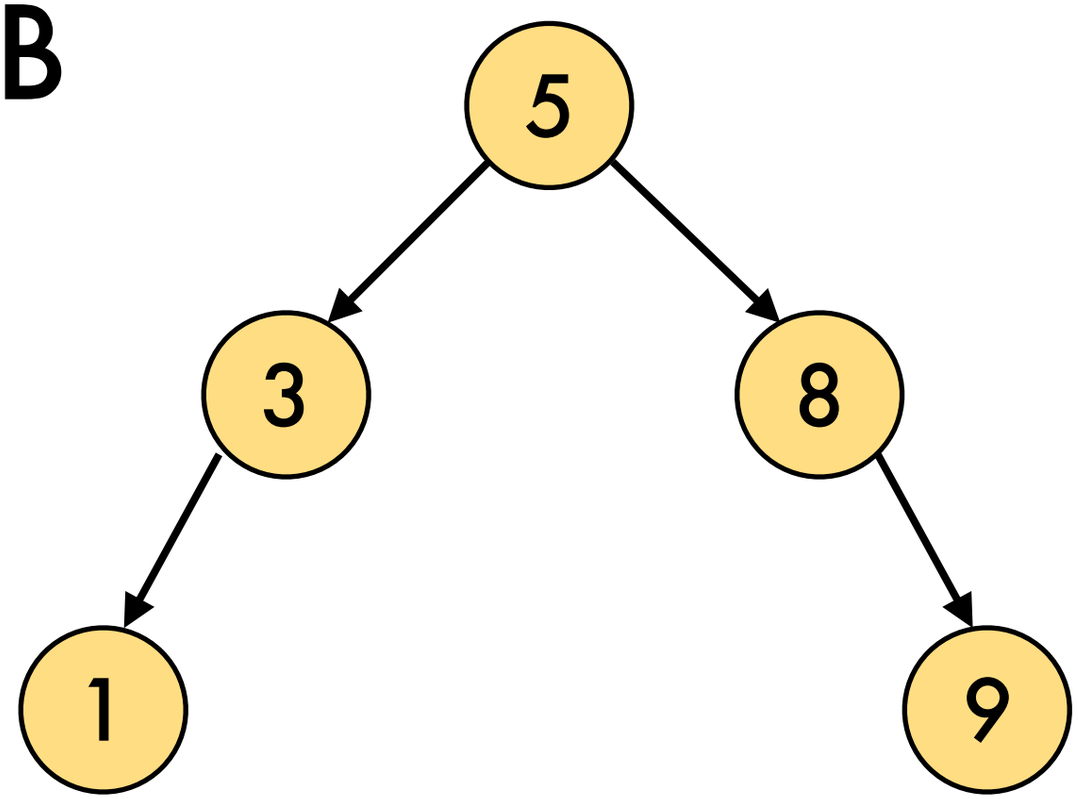
height

$$|h(\triangle) - h(\triangle)| \leq 1$$

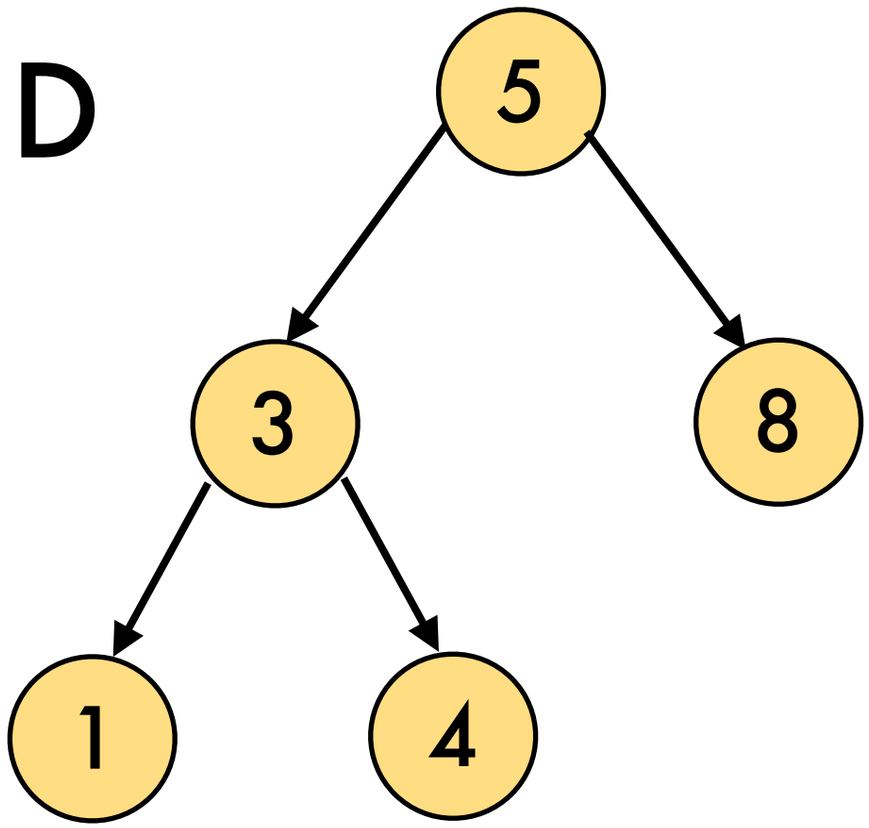
A



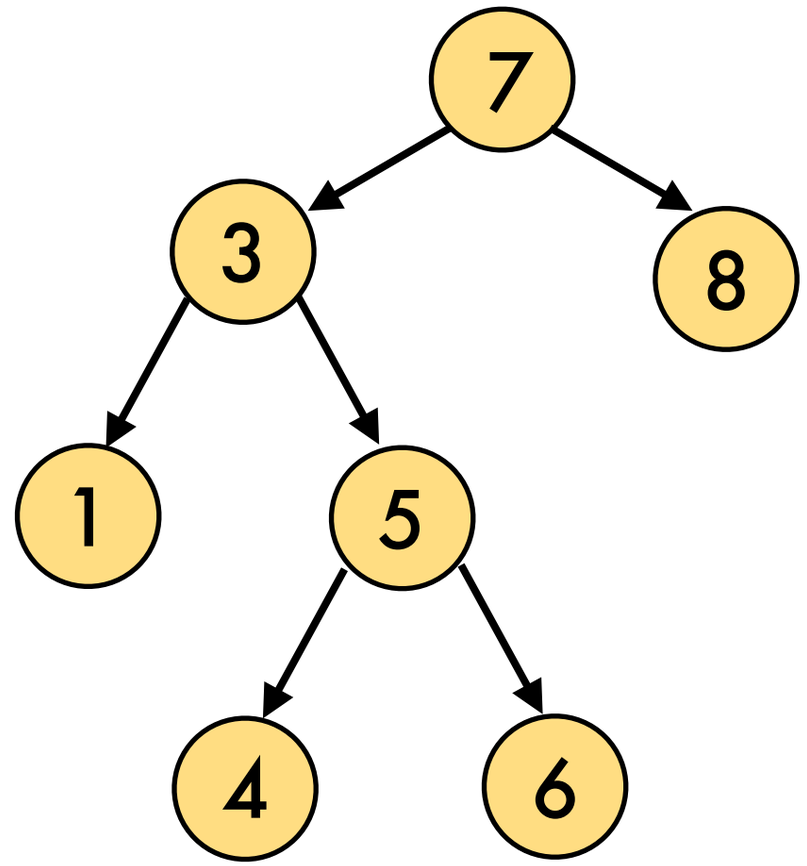
B



D



C

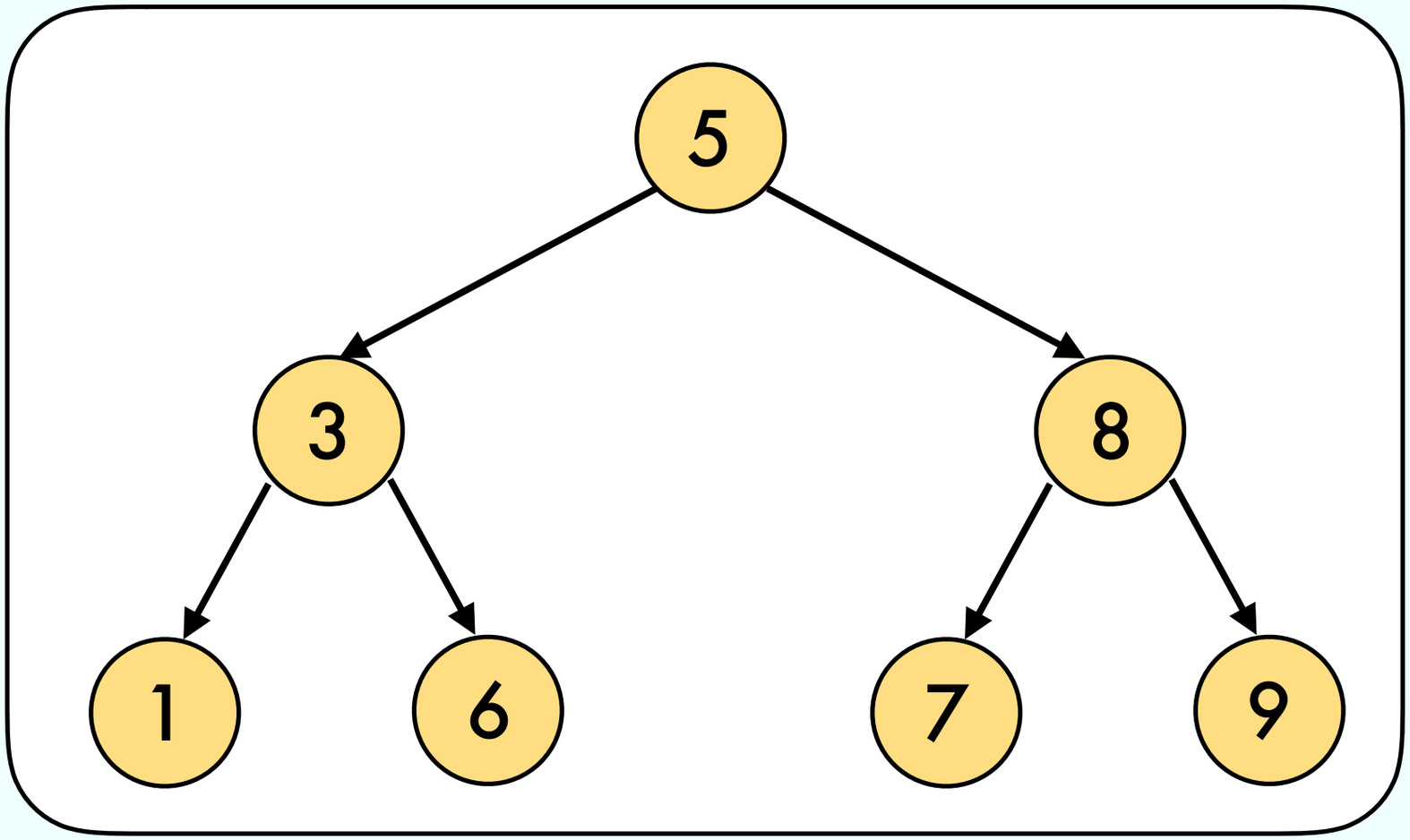


Problem 10

Check if a binary tree is a AVL

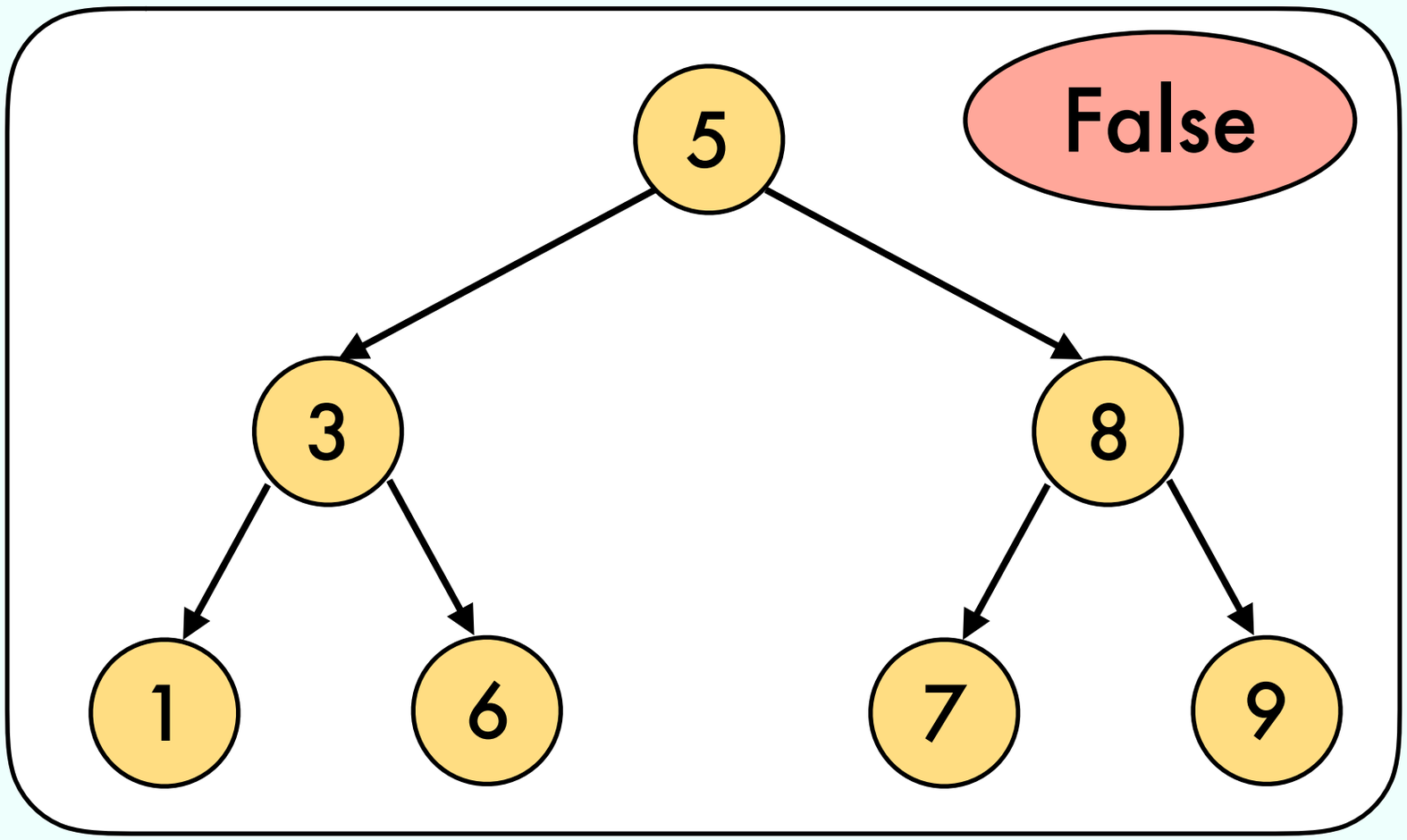
Problem 10

Check if a binary tree is a AVL



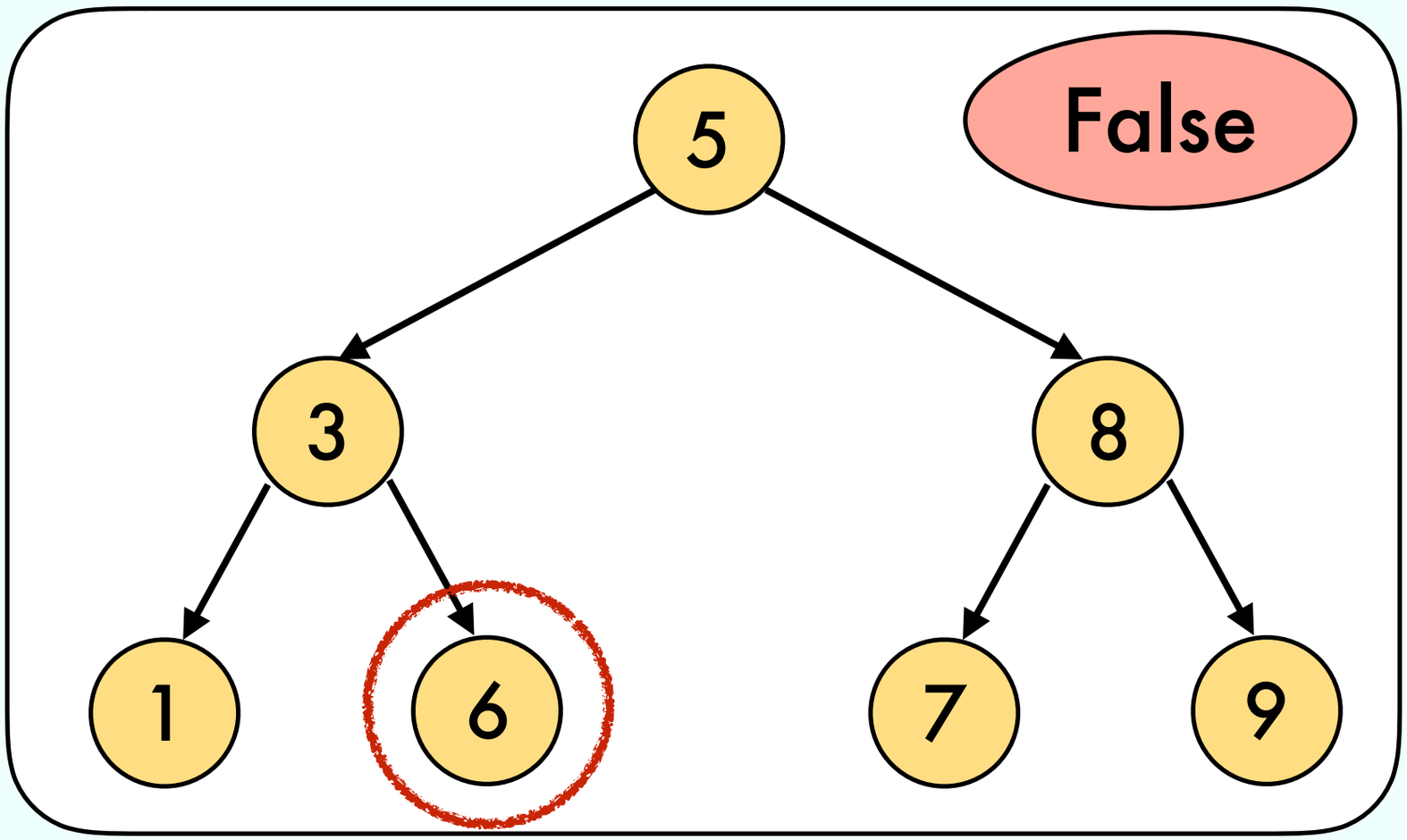
Problem 10

Check if a binary tree is a AVL



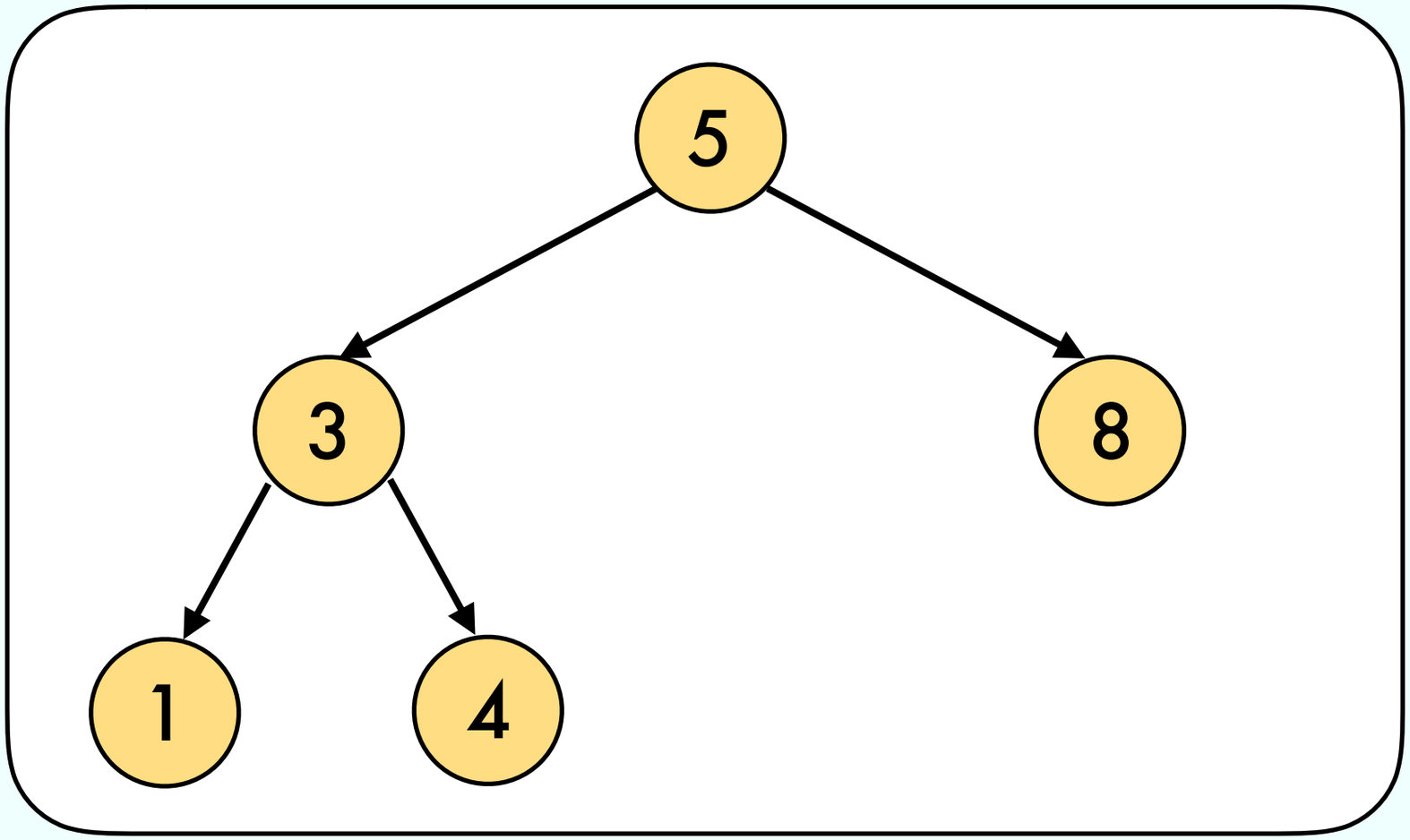
Problem 10

Check if a binary tree is a AVL



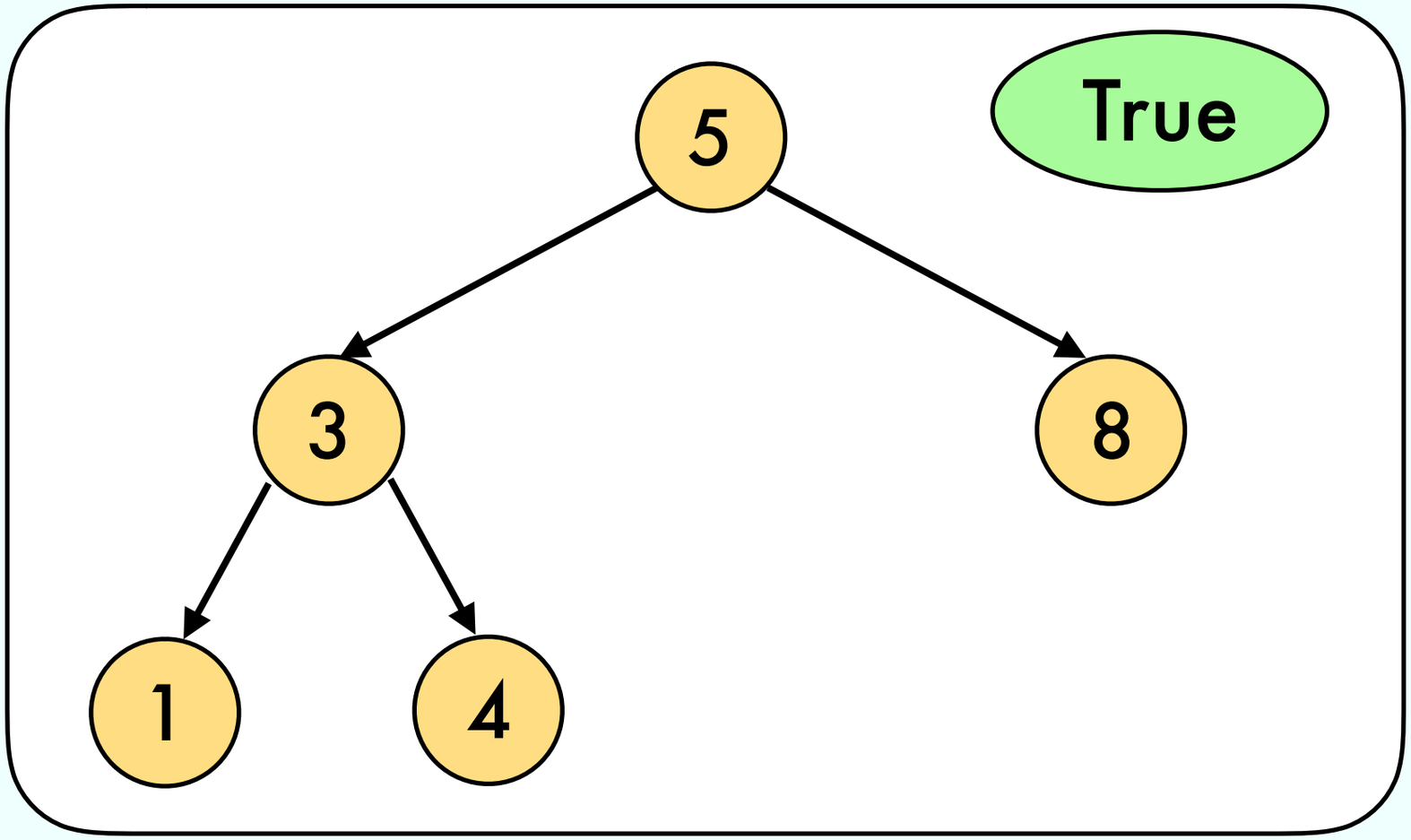
Problem 10

Check if a binary tree is a AVL



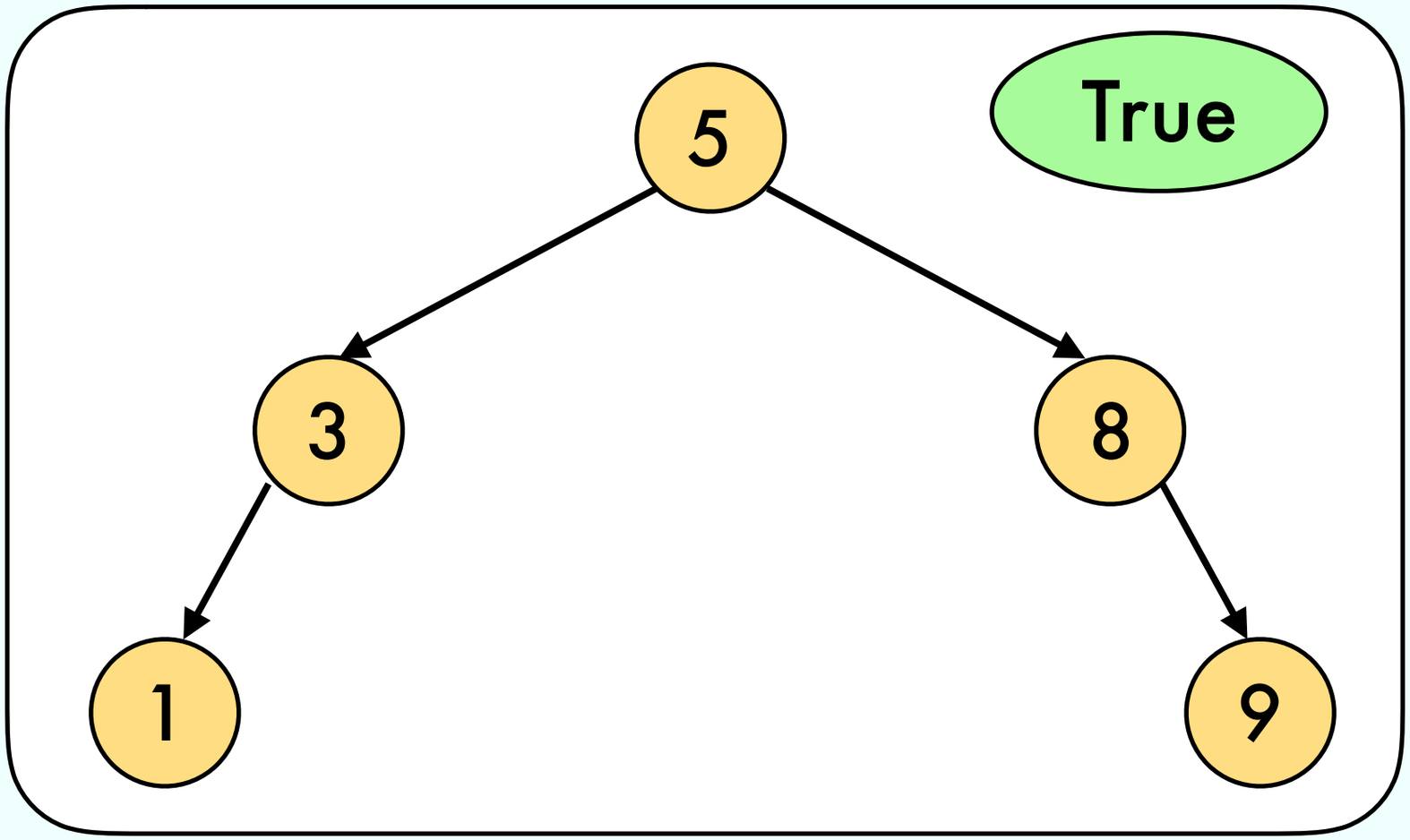
Problem 10

Check if a binary tree is a AVL



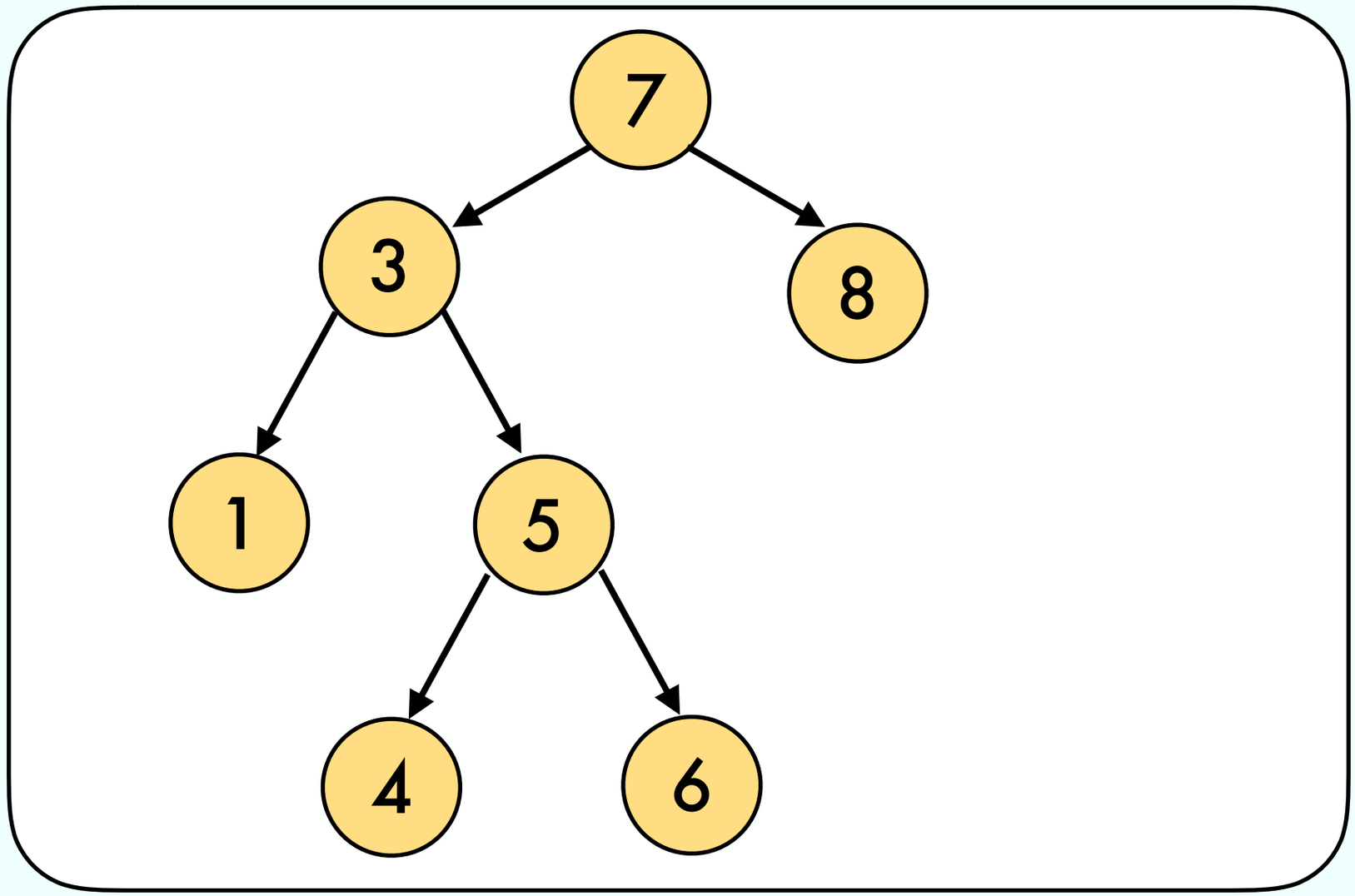
Problem 10

Check if a binary tree is a AVL



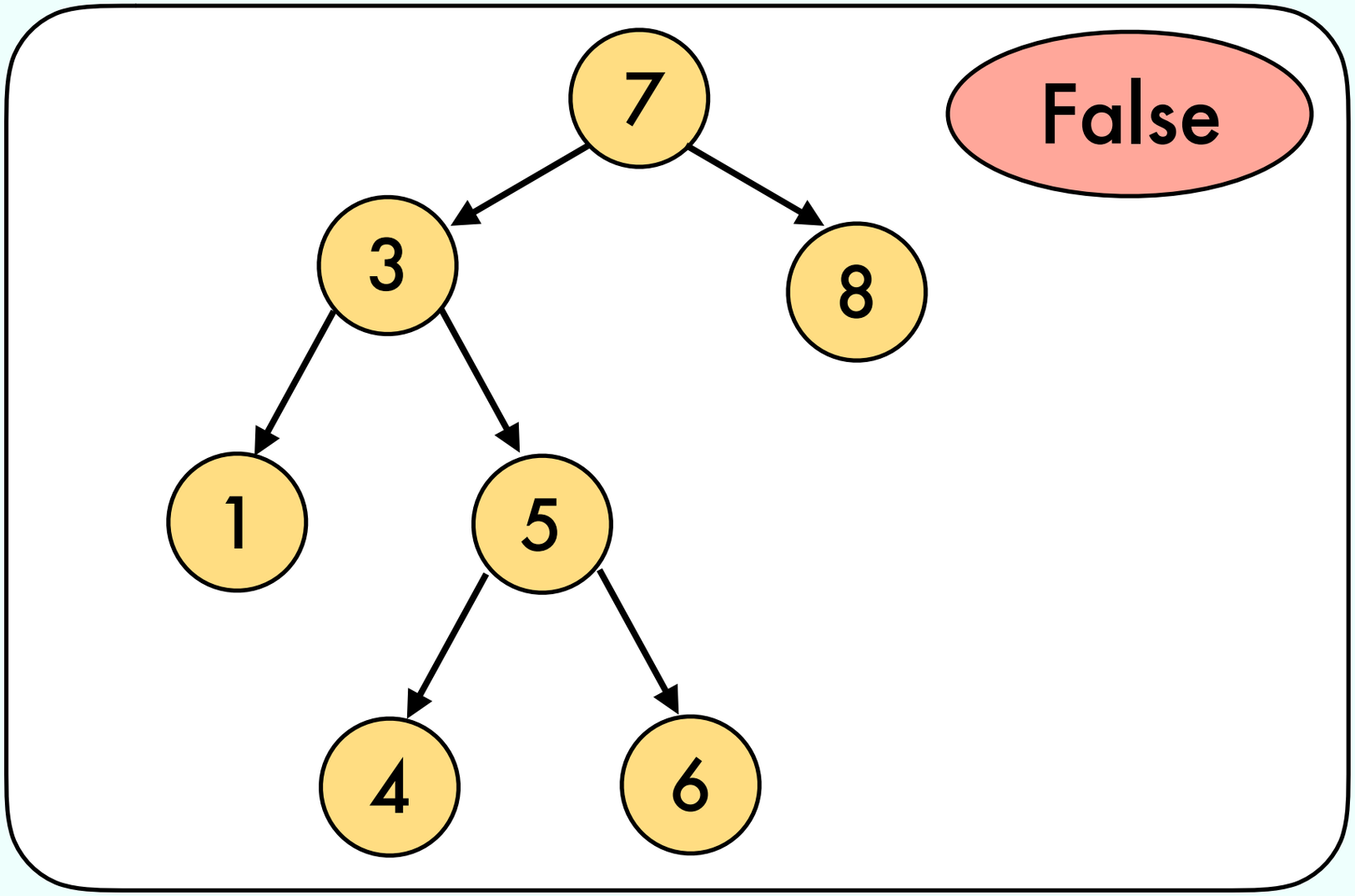
Problem 10

Check if a binary tree is a AVL



Problem 10

Check if a binary tree is a AVL



Problem 10

Check if a binary tree is a AVL

```
Bool isAVL(Tree t) {
 return isBST(t) && isBalanced(t)
 }
```

Problem 10

Check if a binary tree is a AVL

O(N)

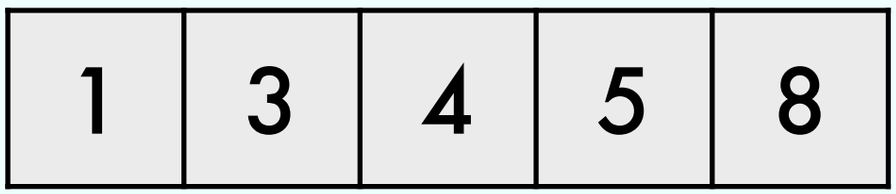
```
Bool isAVL(Tree t) {
 return isBST(t) && isBalanced(t)
 }
```

Exercise 3 from 12/08

Convert a sorted array into a AVL

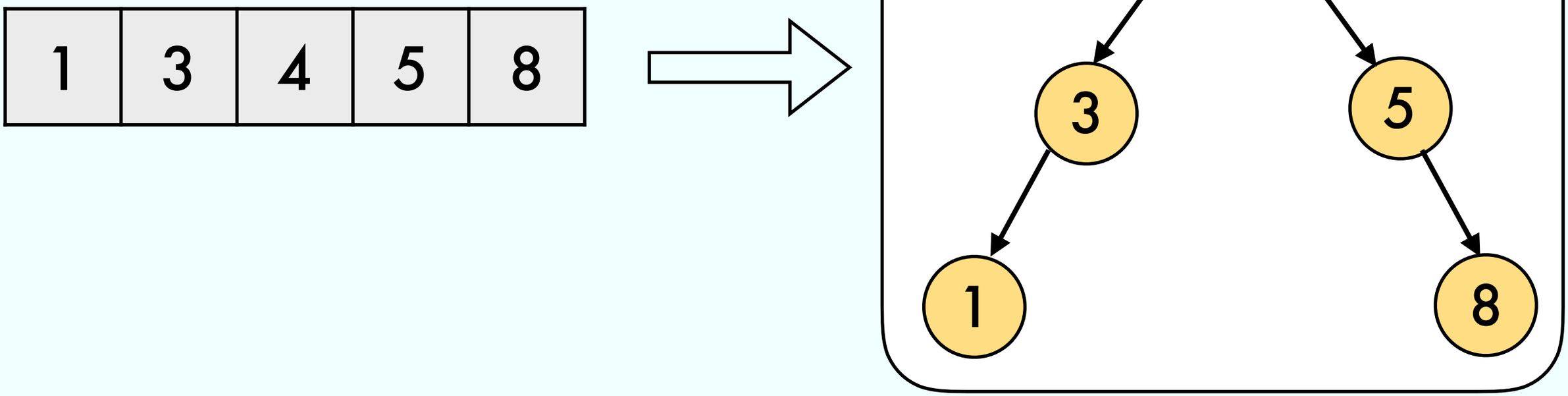
Exercise 3 from 12/08

Convert a sorted array into a AVL



Exercise 3 from 12/08

Convert a sorted array into a AVL

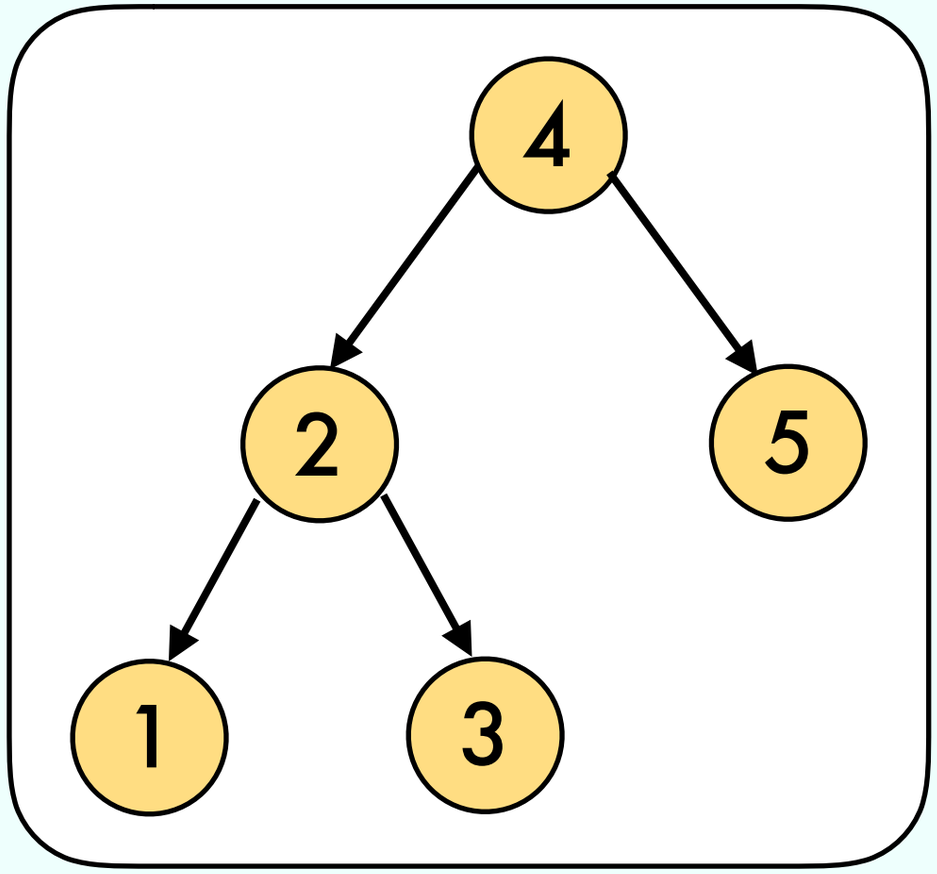


Exercise 4 from 12/12

Find the n-th element in an AVL

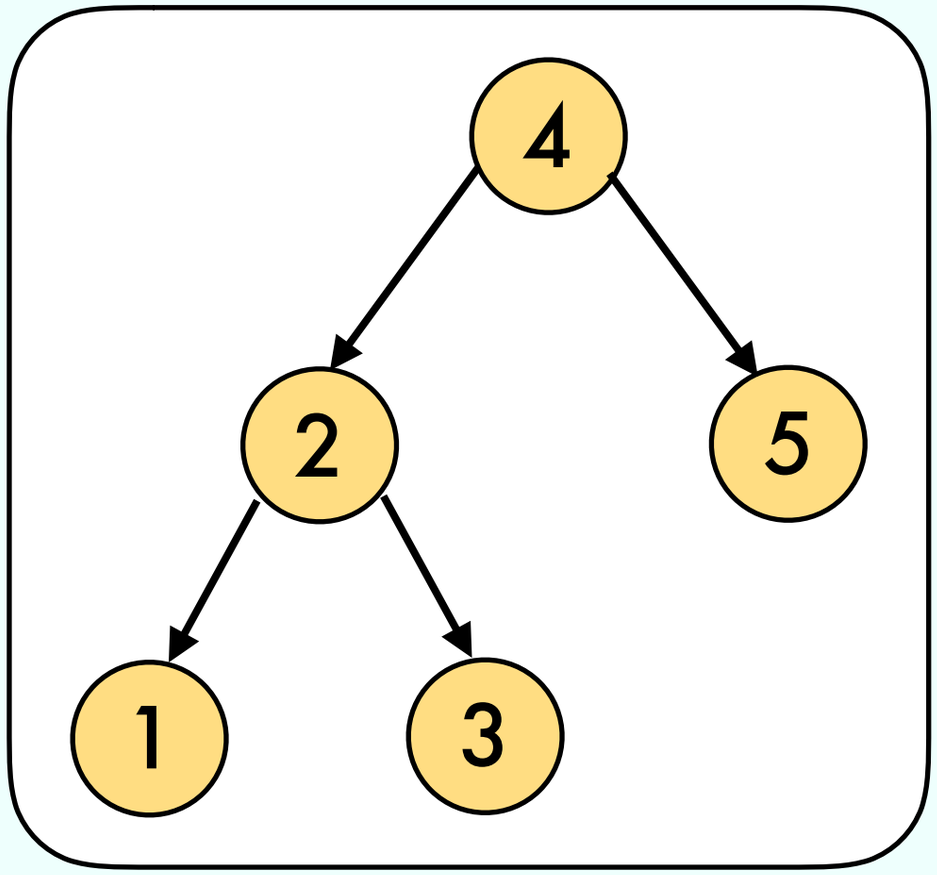
Exercise 4 from 12/12

Find the n-th element in an AVL

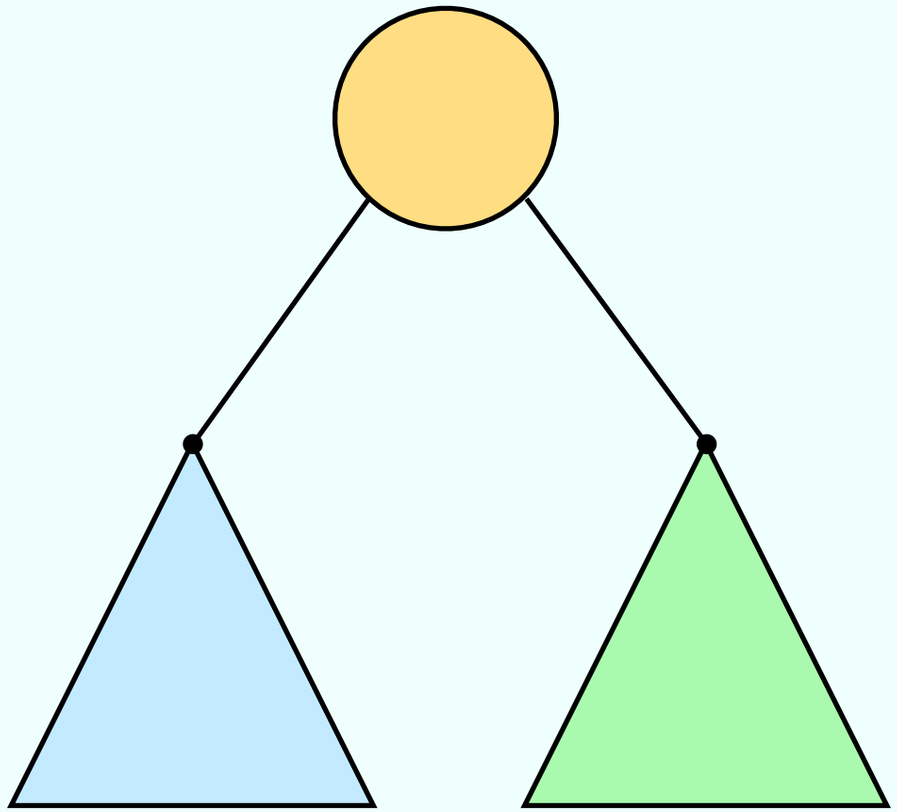


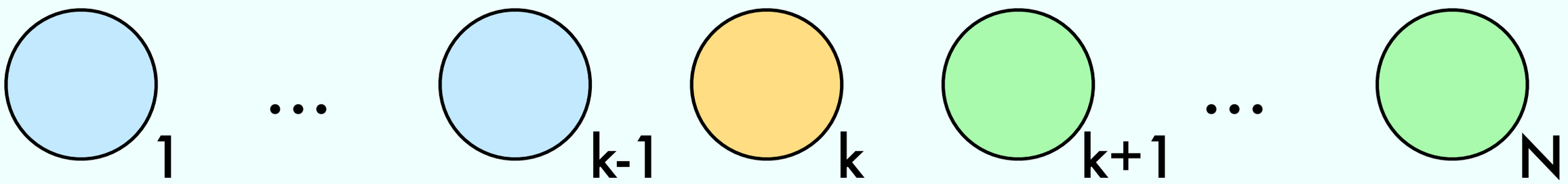
Exercise 4 from 12/12

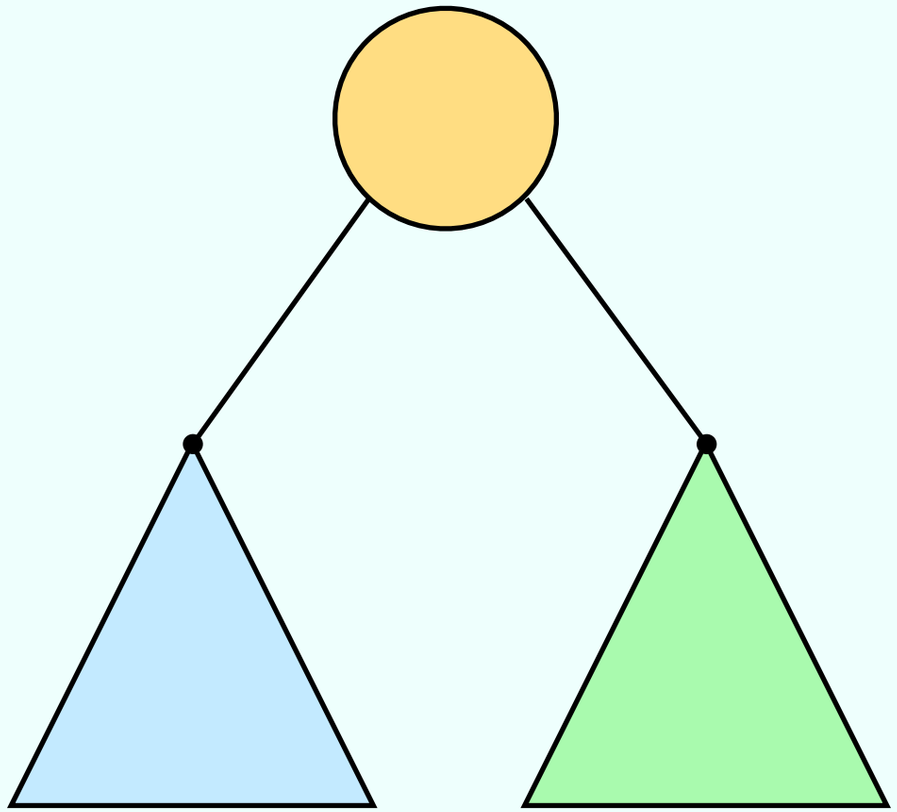
Find the n-th element in an AVL

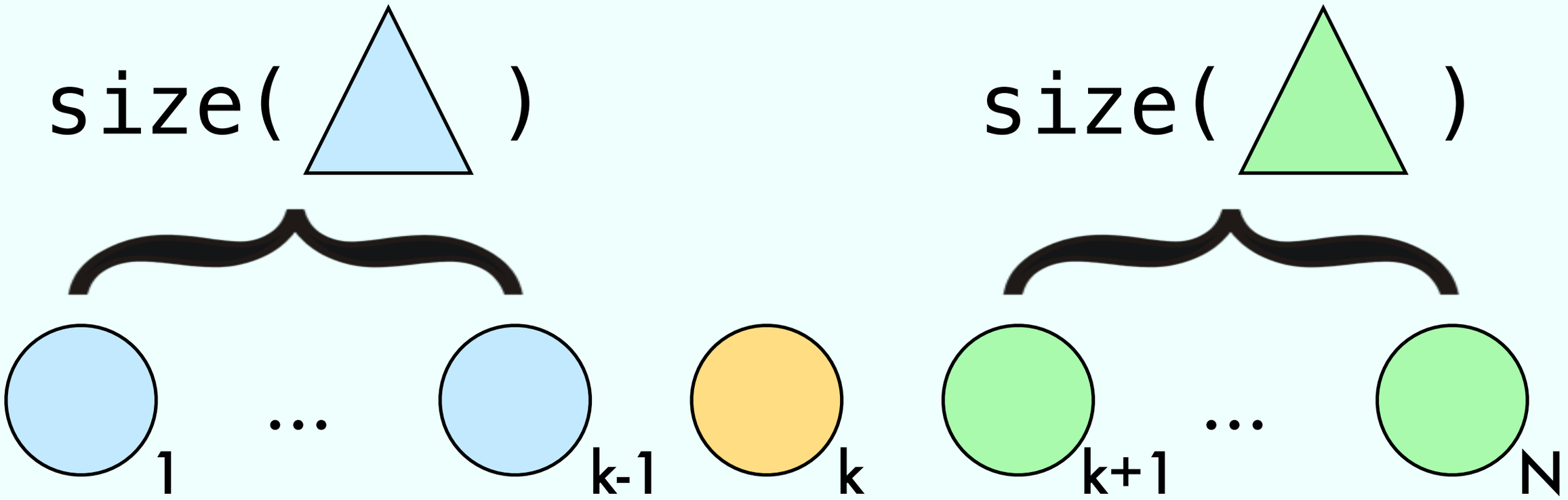



```
nth-element(1) = 1

nth-element(2) = 2

nth-element(3) = 3

nth-element(4) = 4

nth-element(5) = 5
```




1
...
k-1
k
k+1
...
N




size(
)
1
...
k-1
k
size(
)
k+1
...
N