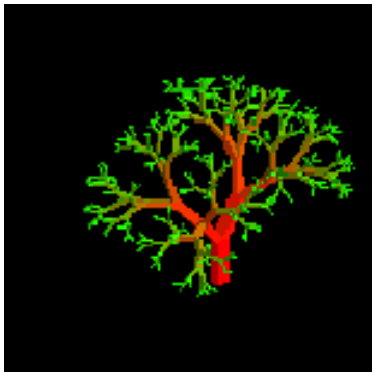




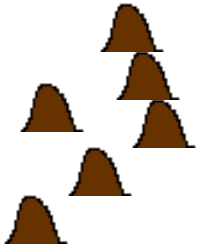




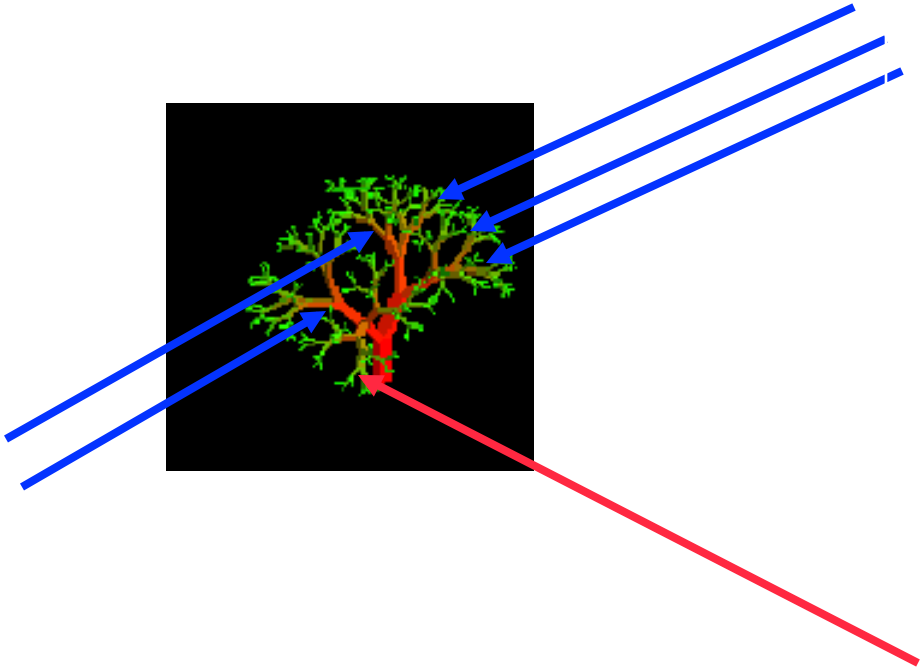


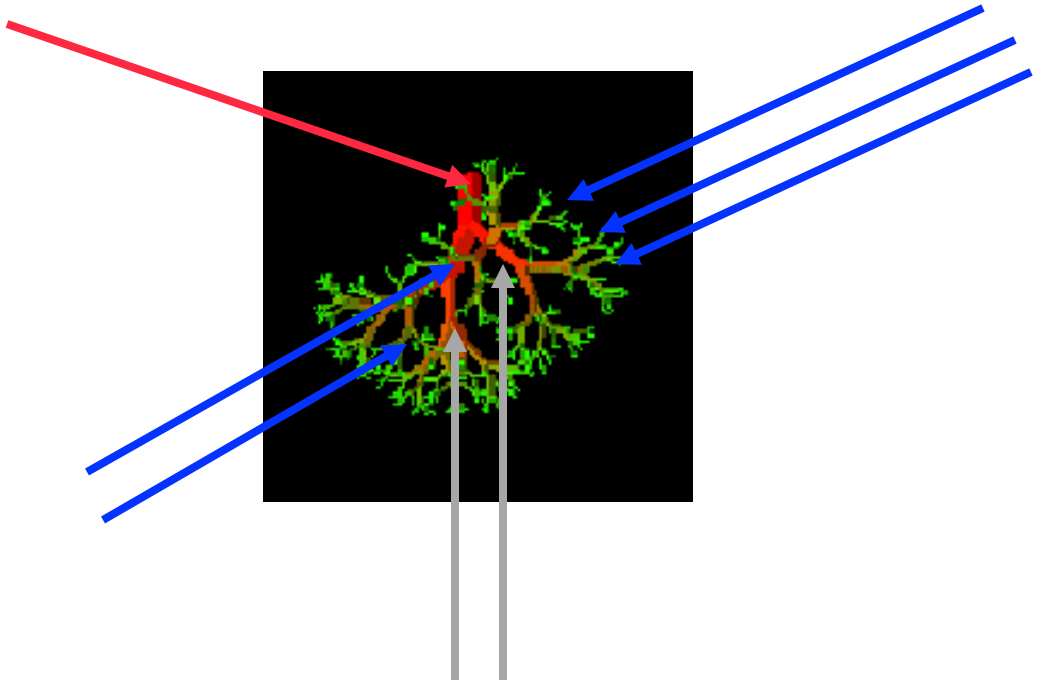








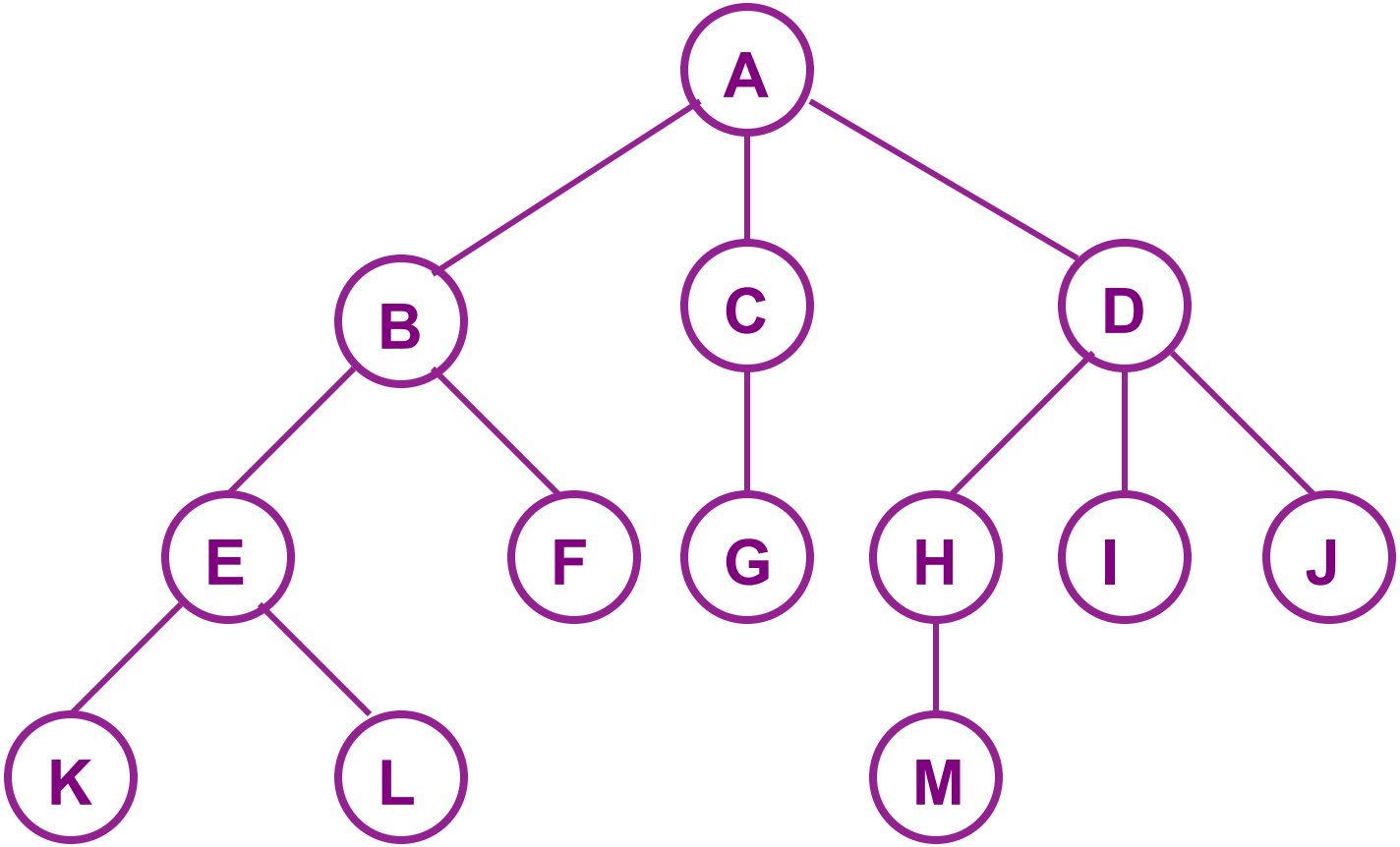




Chapter 5

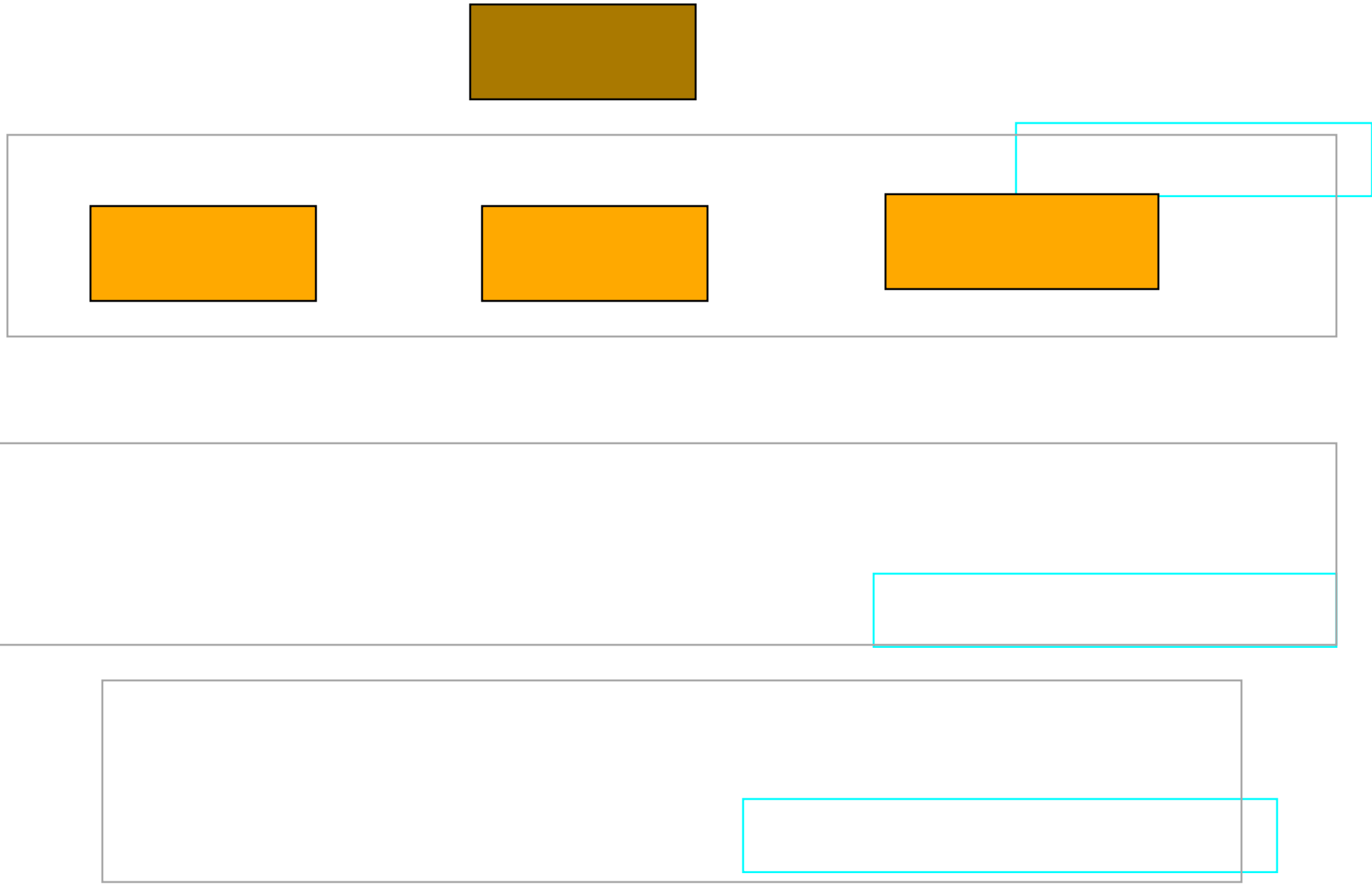
Trees

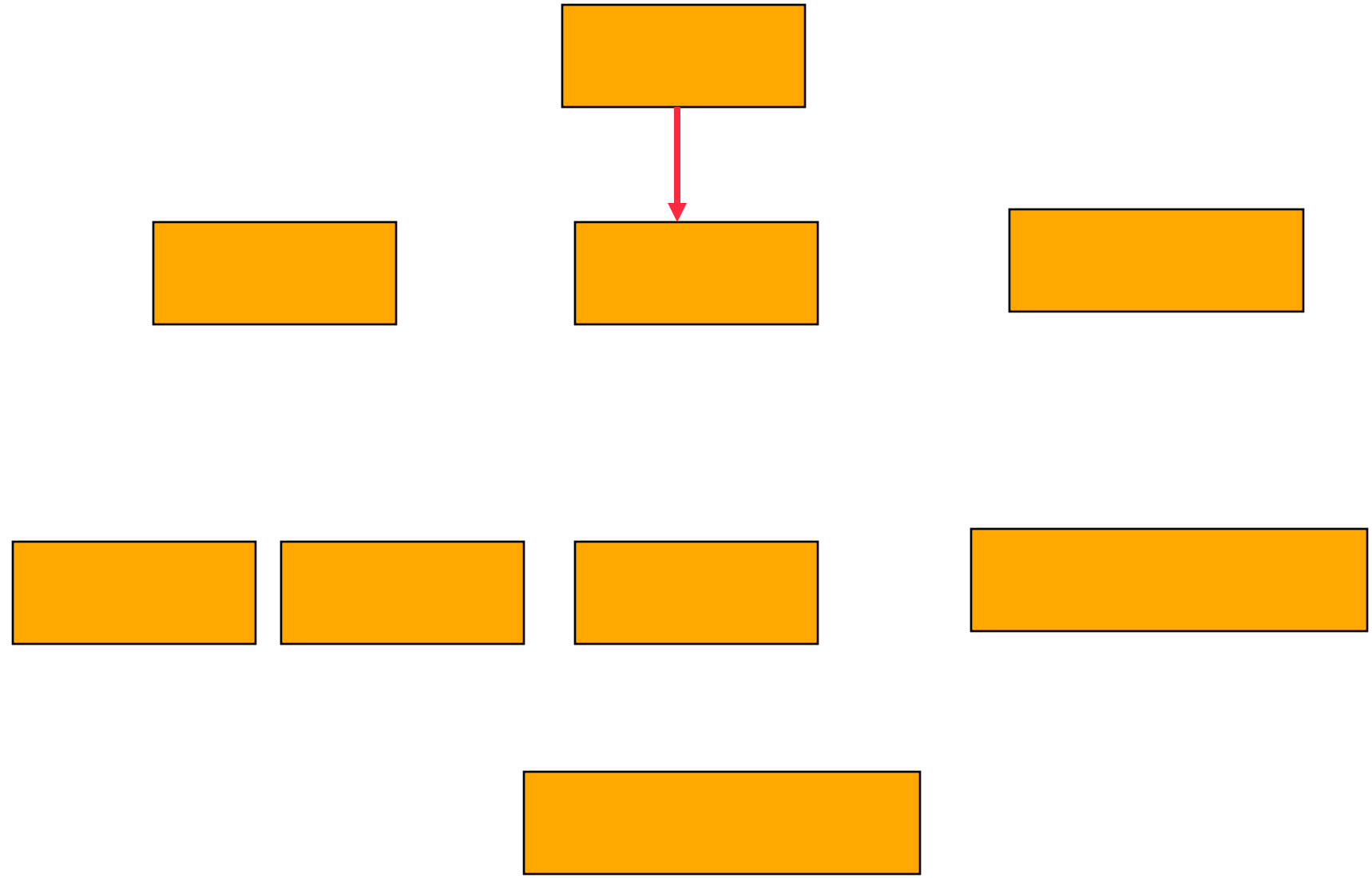
≥

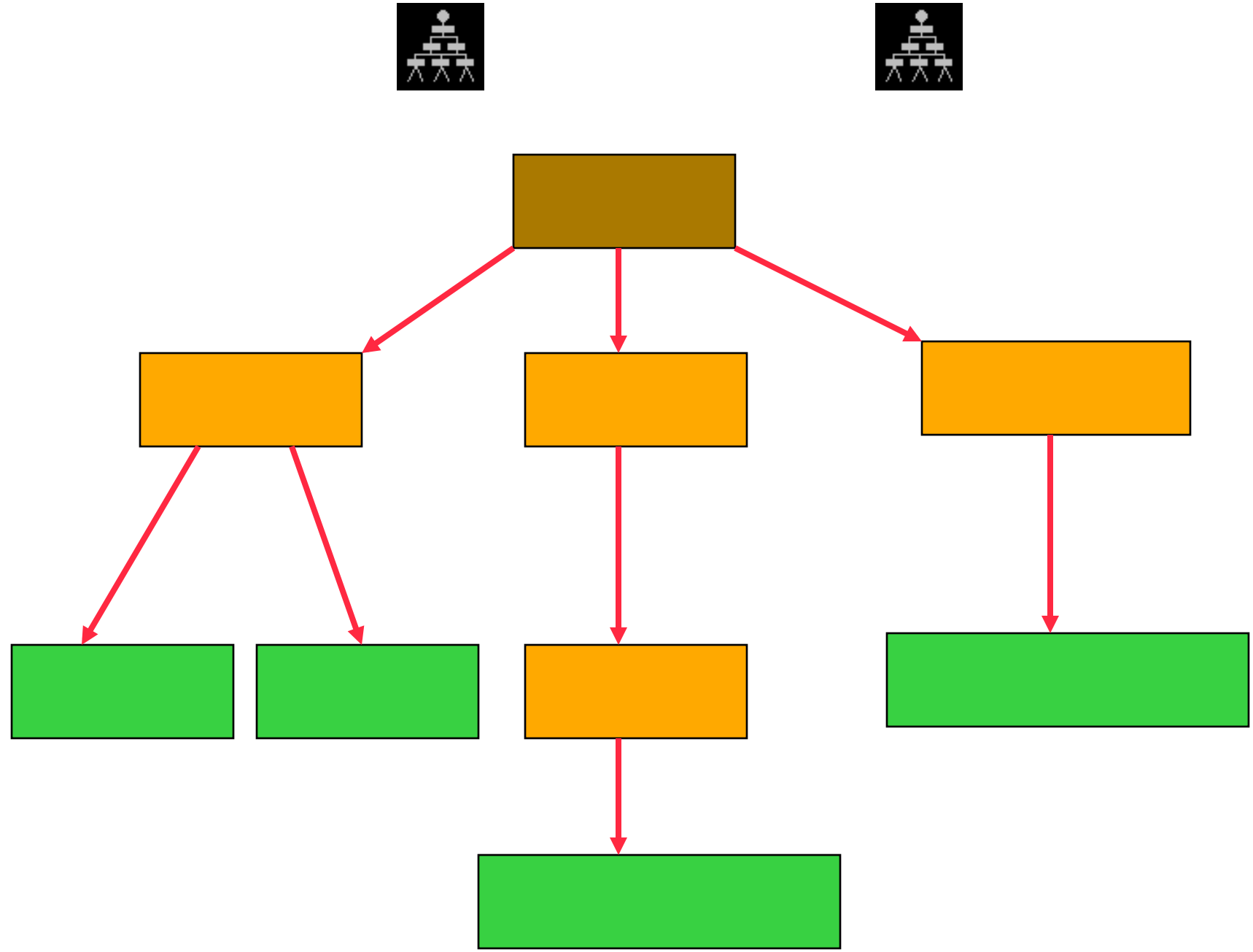

A
B
C
D
E
F
G
H
I
J
K
L
M

- (0, 1, 2, , -1)
-
-
-
-
-

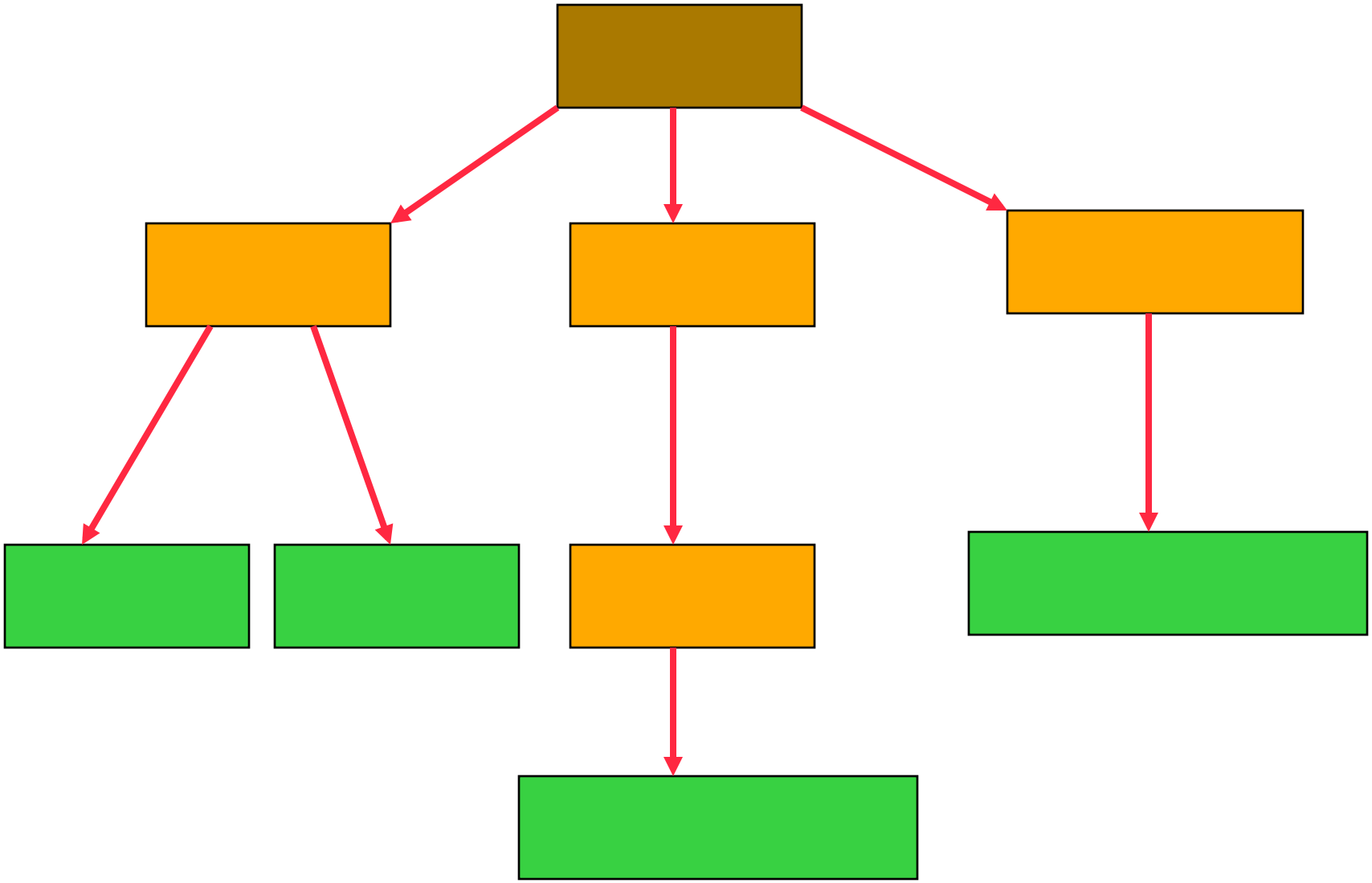
(1.1)

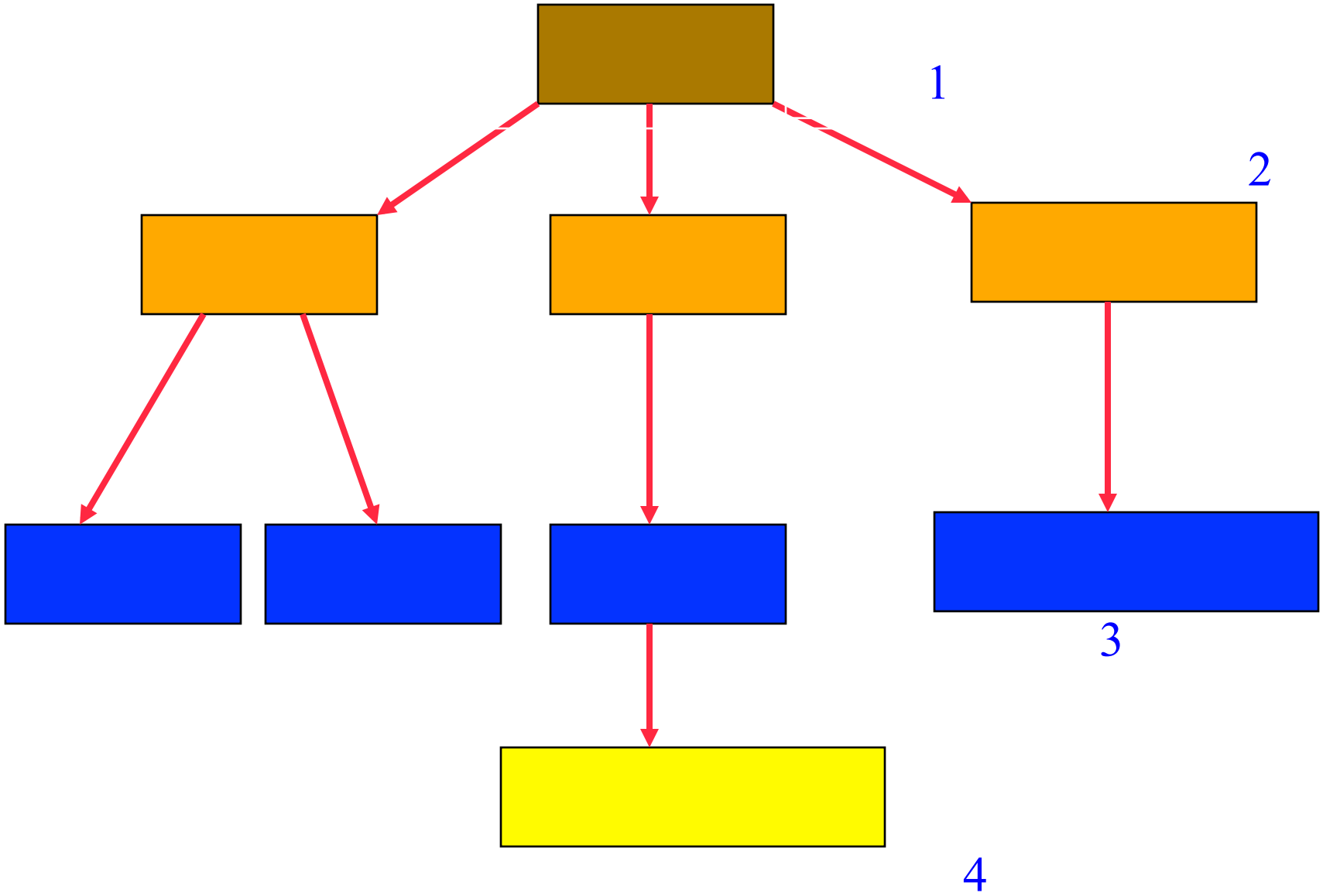






, , , ,




1
2
3
4

0

1.

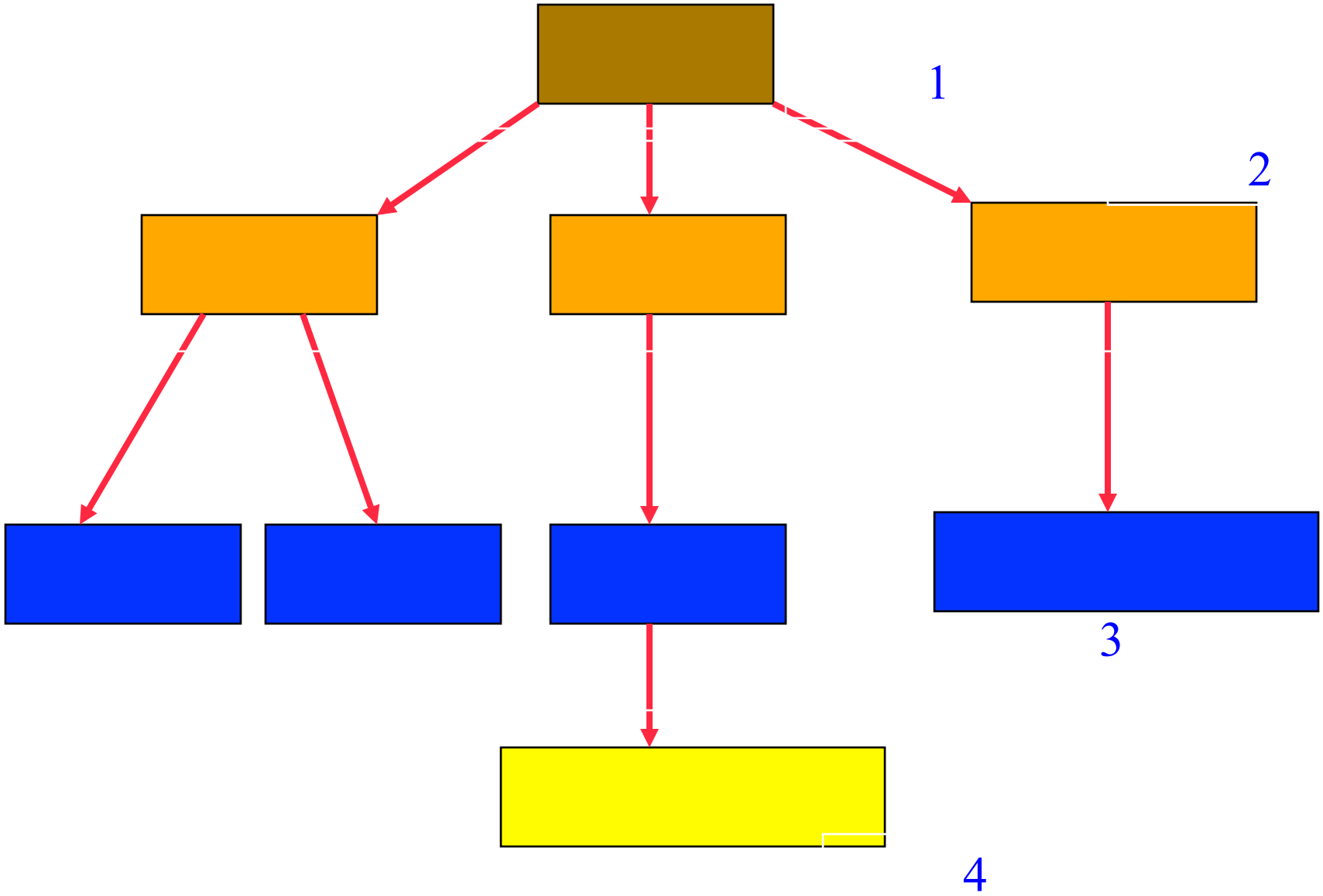
0.

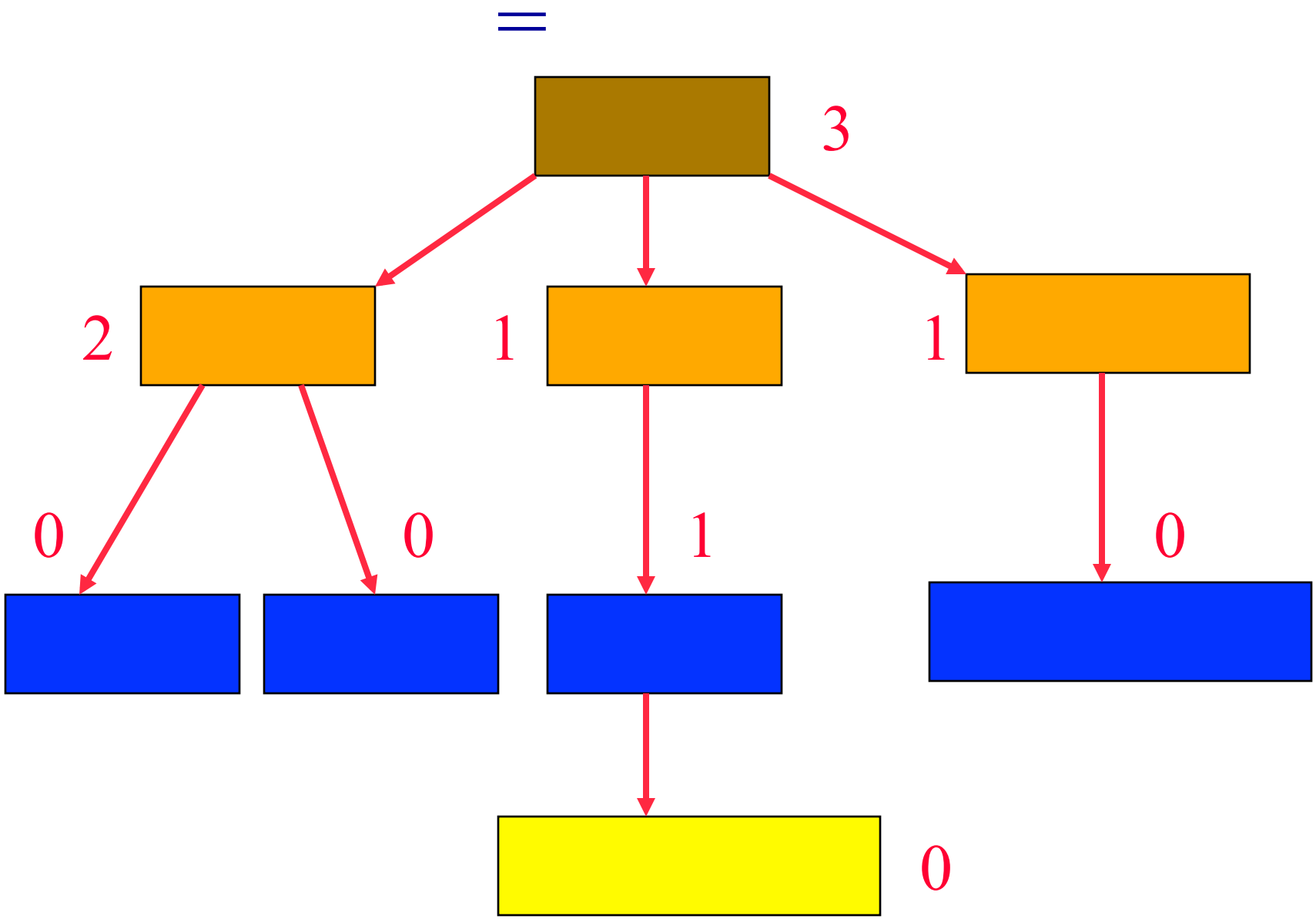
1.

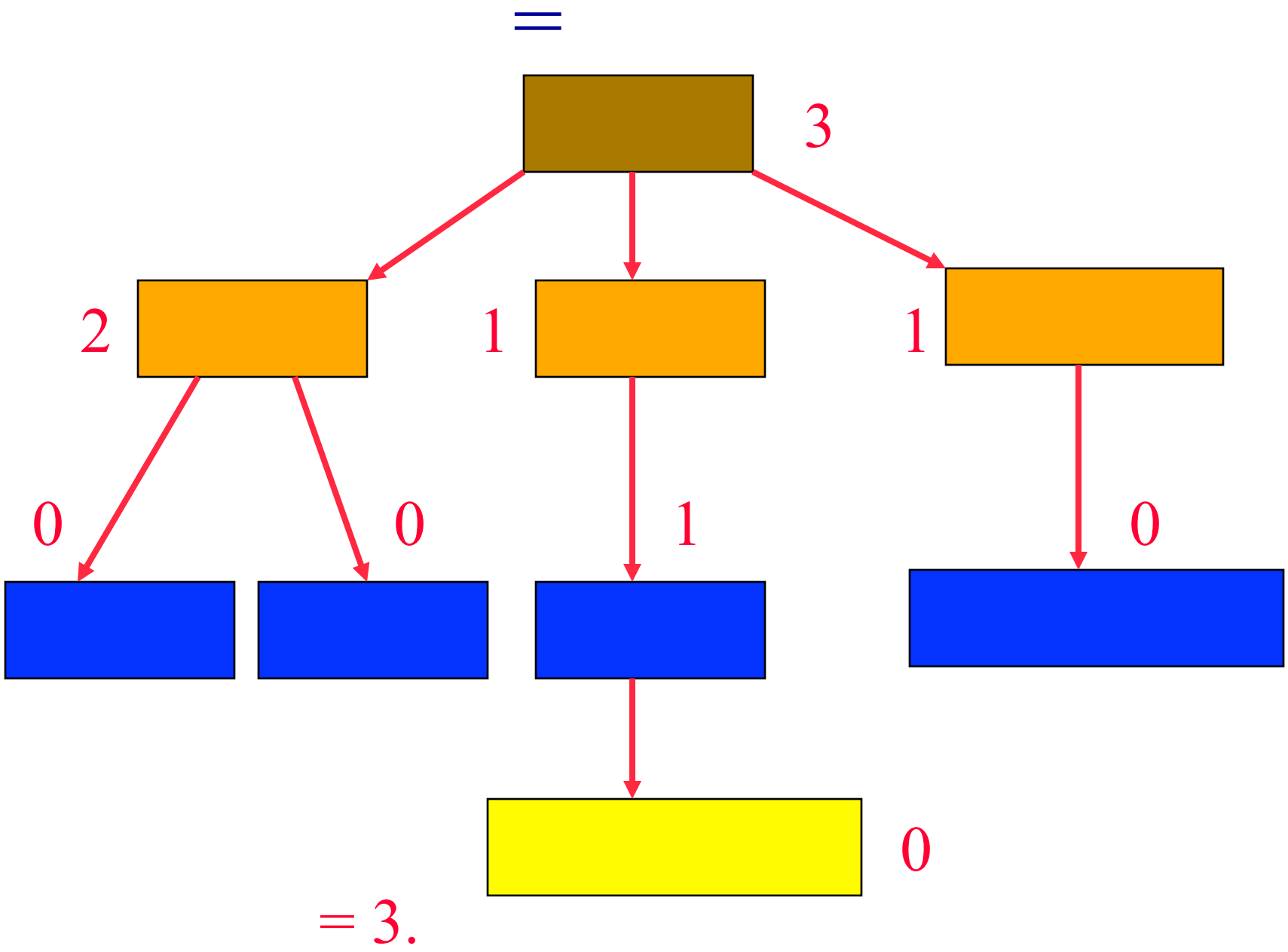
2.

.

1.

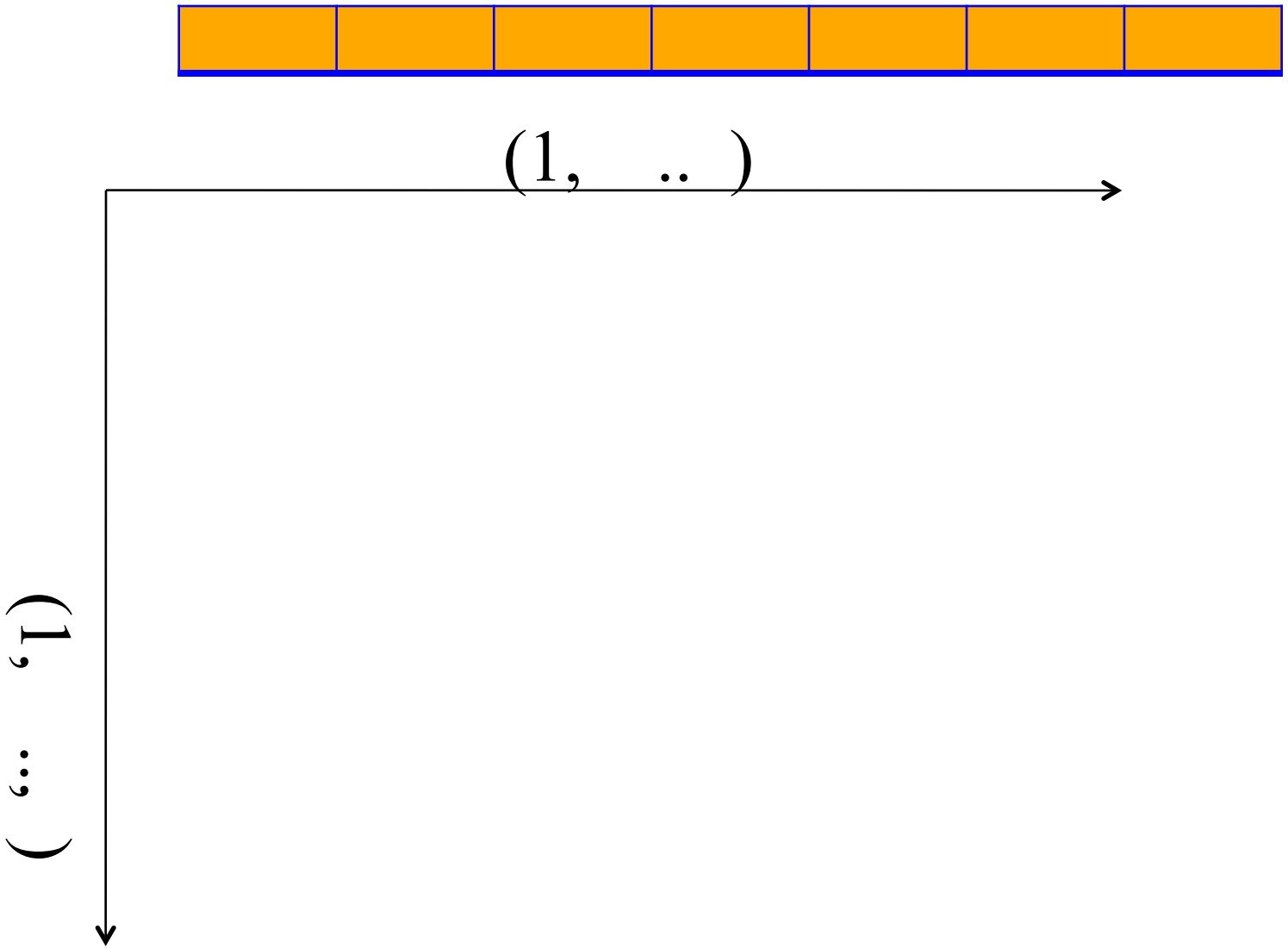

1
2
3
4


=
3
2
1
1
0
0
1
0
0


=
3
2
1
1
0
0
1
0
0
= 3.

-
- ( )

5.1.2 Representation of Trees



5.1.2 Representation of Trees

,

Data	Child1	Child2		Child k

Possible node structure for a tree c

≥

( ) .

.

( )

.

.

&

2,

.

;

.

&

ADT 5.1

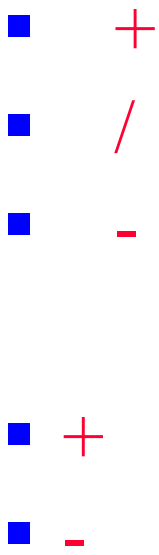
```
template <class T>
class BinaryTree
{ // A finite set of nodes either empty or consisting of
  // a root node, left BinaryTree and right BinaryTree.
public:
    BinaryTree ();
    // creates an empty binary tree

    bool IsEmpty ();
    // return true iff the binary tree is empty

    BinaryTree(BinaryTree<T>& bt1, T& item,


BinaryTree<T>& bt2);
    // creates a binary tree whose left subtree is bt1,
    // right subtree is bt2, and root node contain item.
```

```
    BinaryTree  LeftSubtree();
    // return the left subtree of *this

    T RootData();
    // return the data in the root of *this

    BinaryTree  RightSutree();
    // return the right subtree of *this
};
```

(+) * (+) + / * + 3.25

.

- (+, -, /, *).
- (, , , , , , , , 3.25, (+), (+), .).
- ((,)).



- + *
- * + /

.

- (*) = (/) > (+) = (-)

,

.

,

.

- \+ -
- * / /

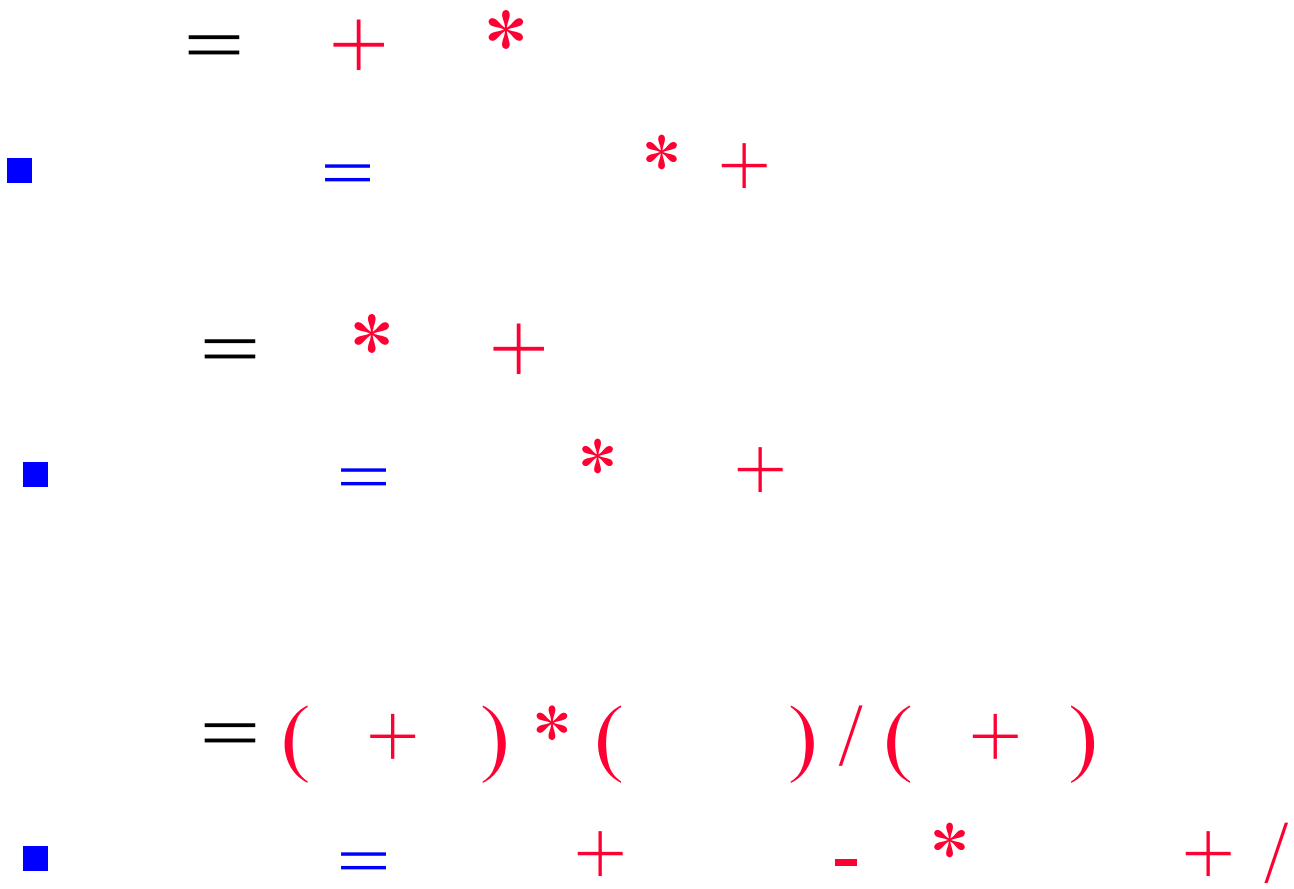
,

.

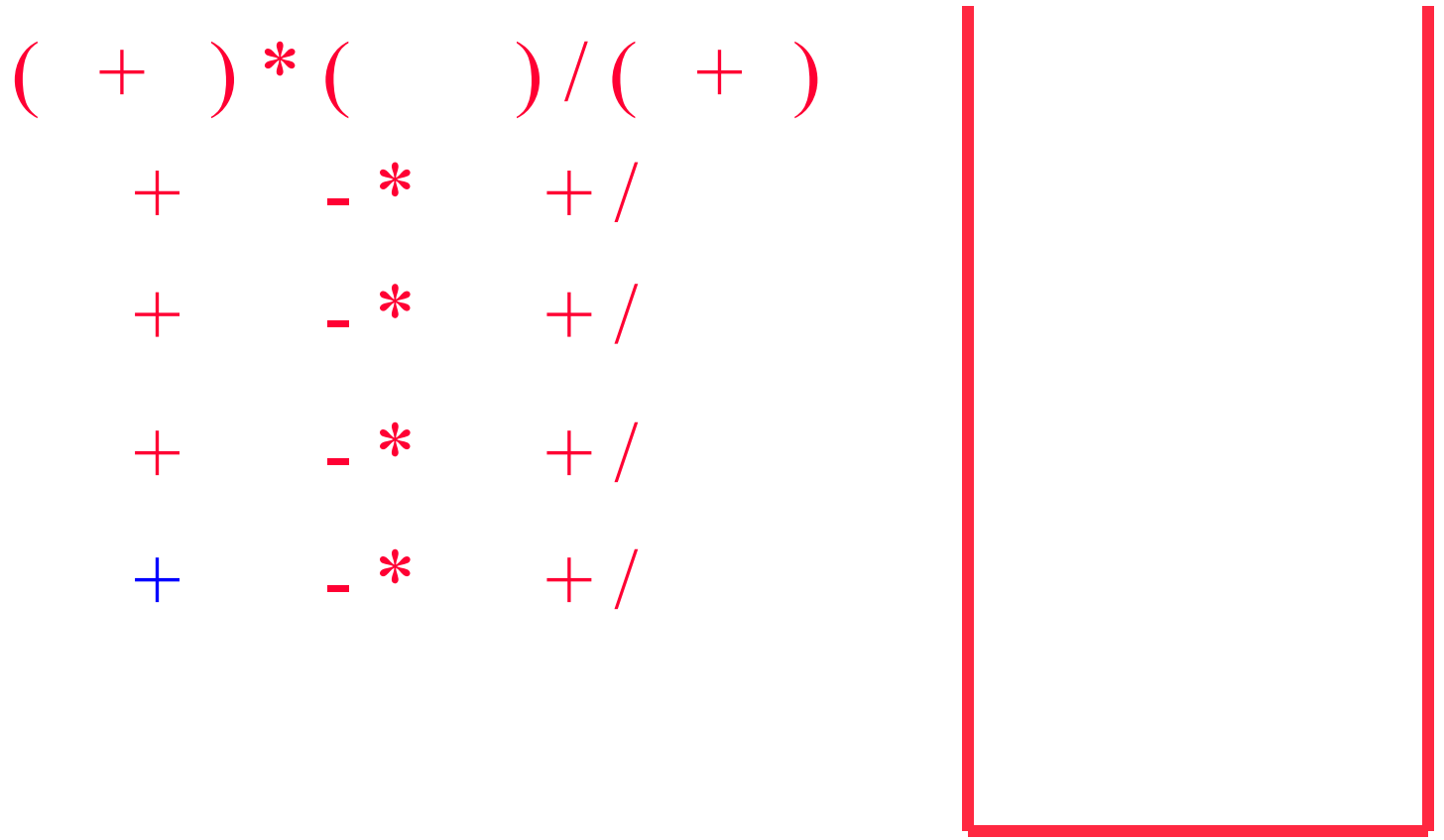
- (+) * ( ) / ( )

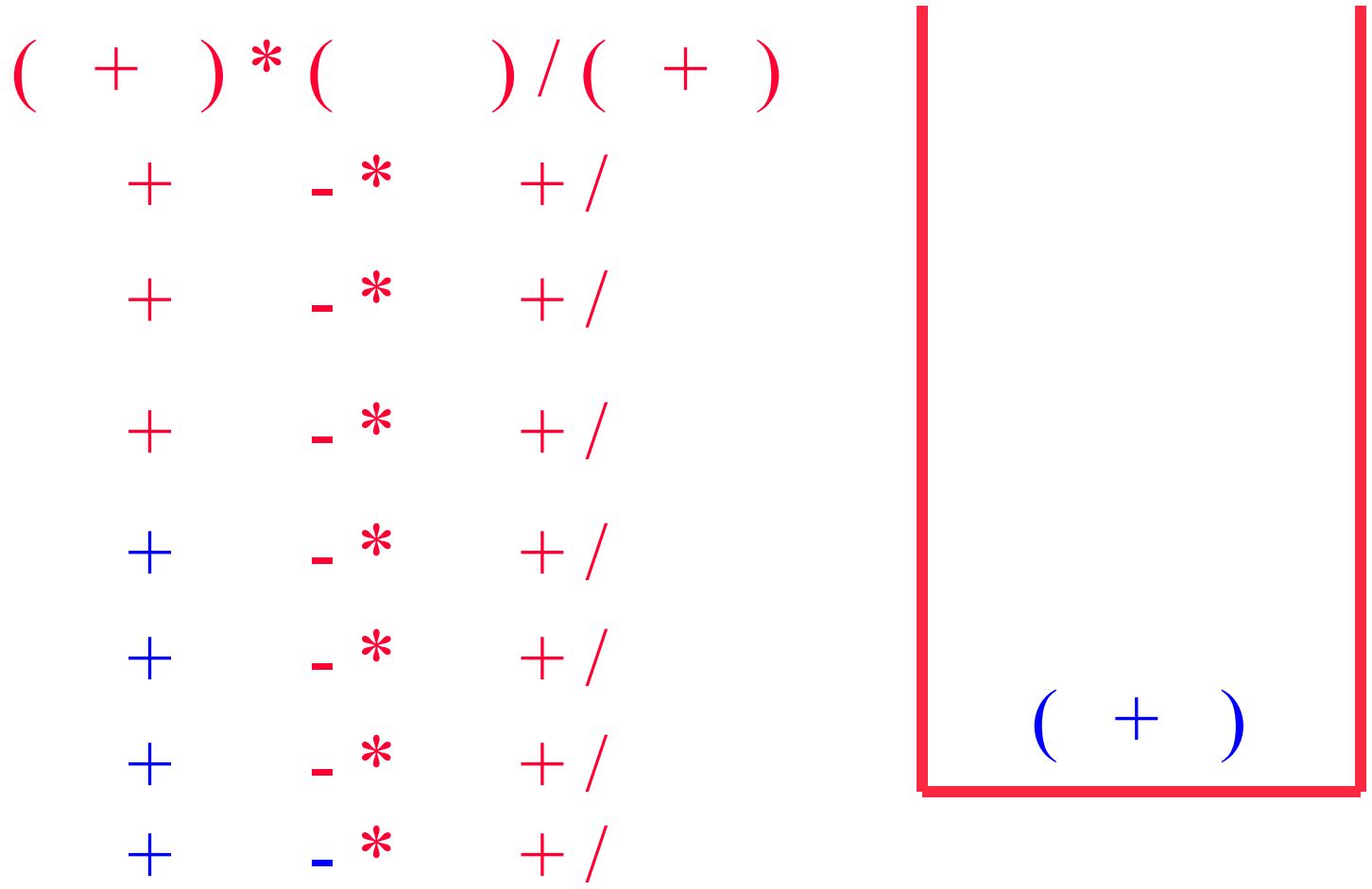
- , , 3.25

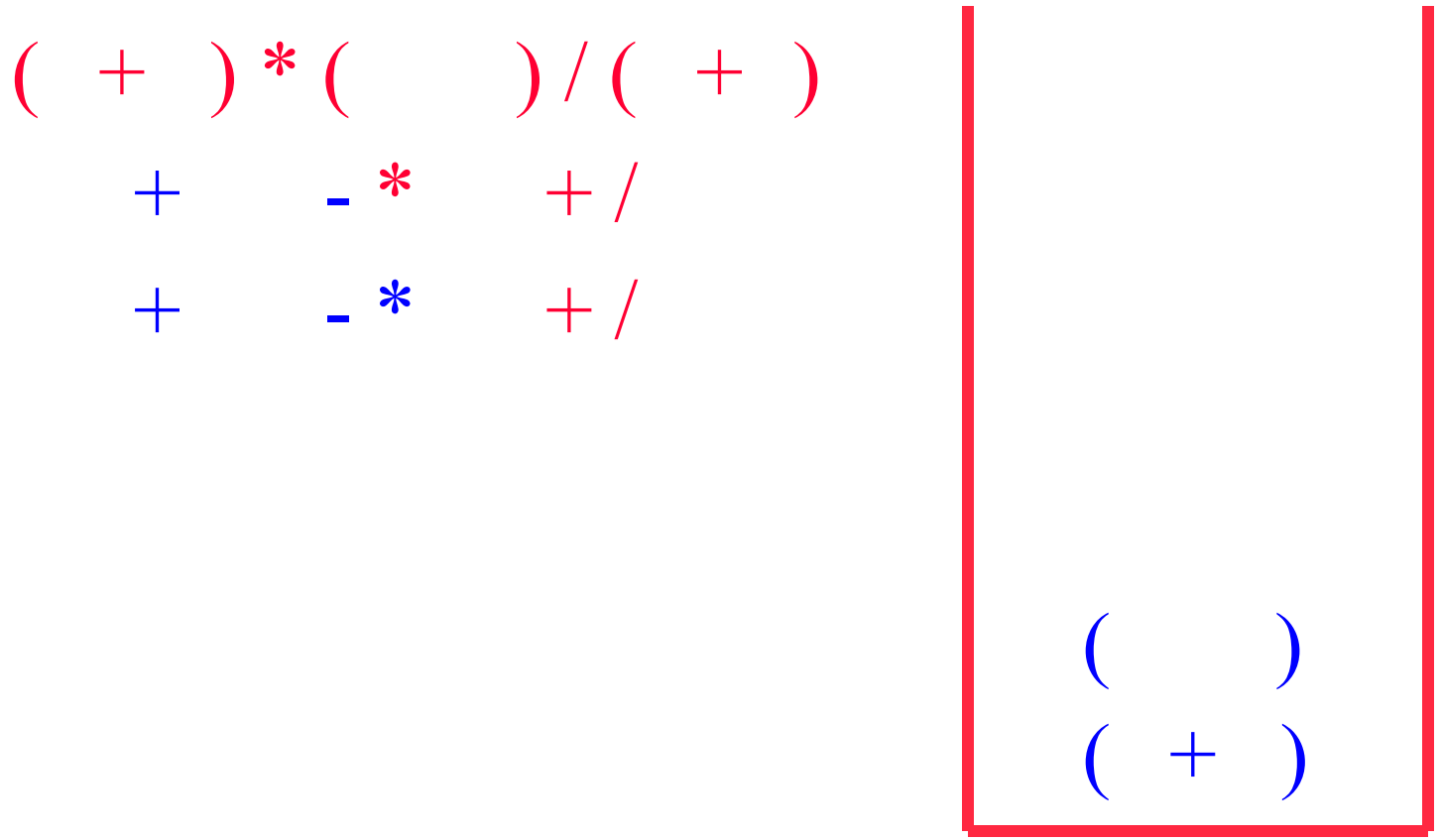
- = +
- = +

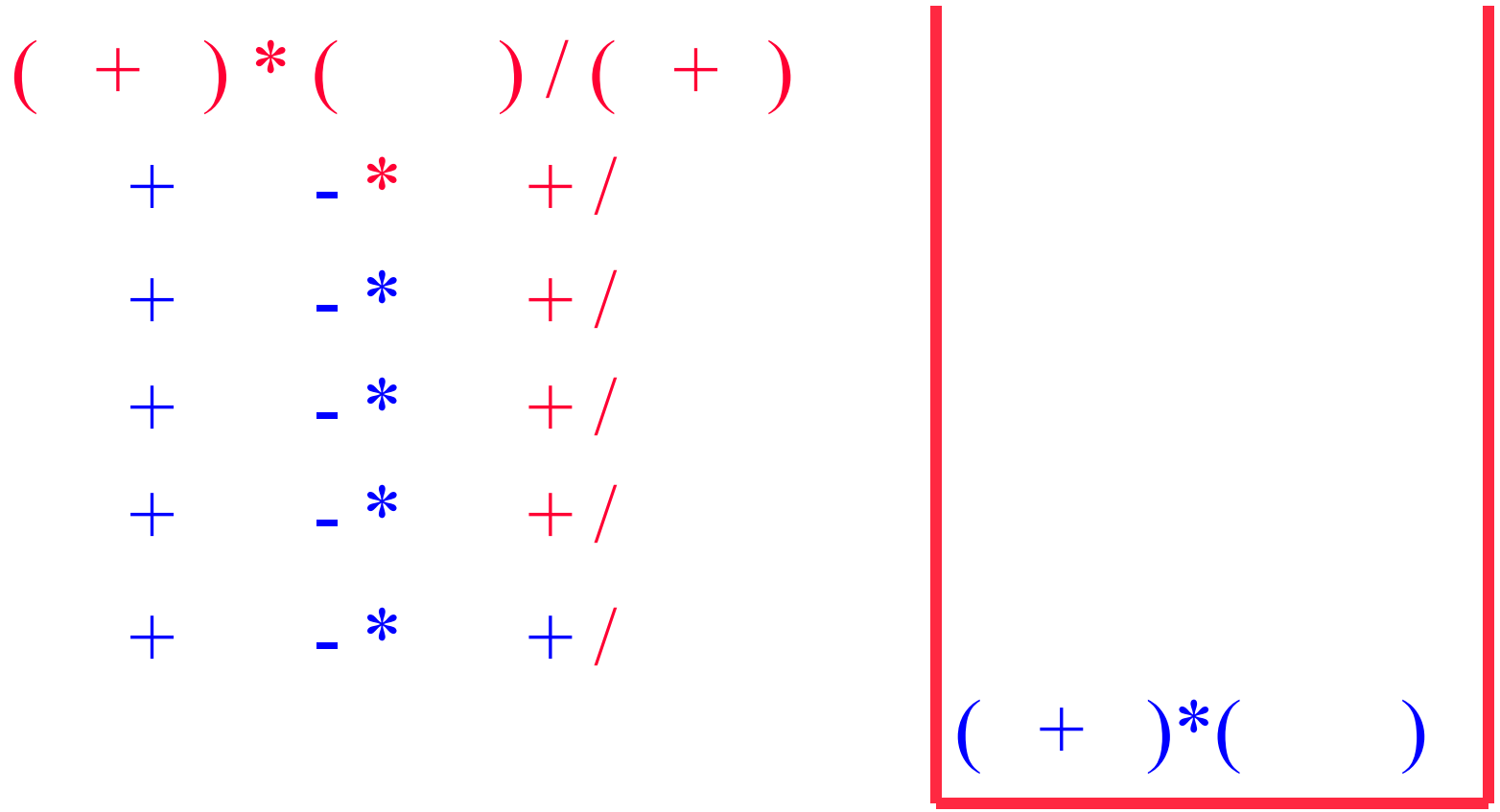


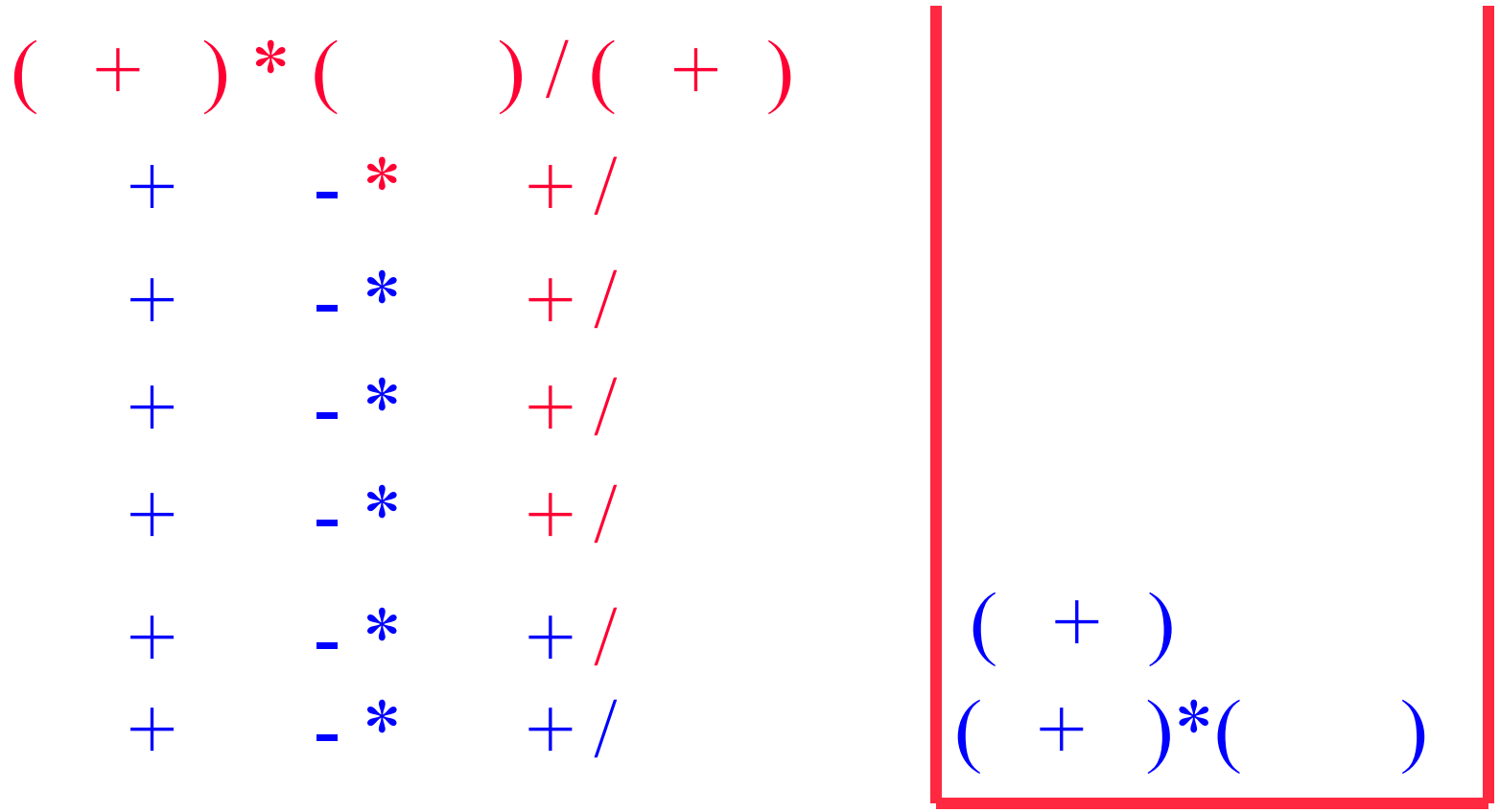
- + =>
- + + => +
- - =>
- - - => -







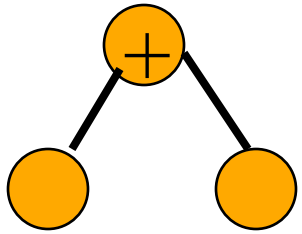


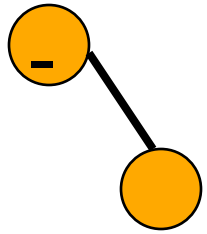


- , , 3.25
- = +
- = +
- = +

+

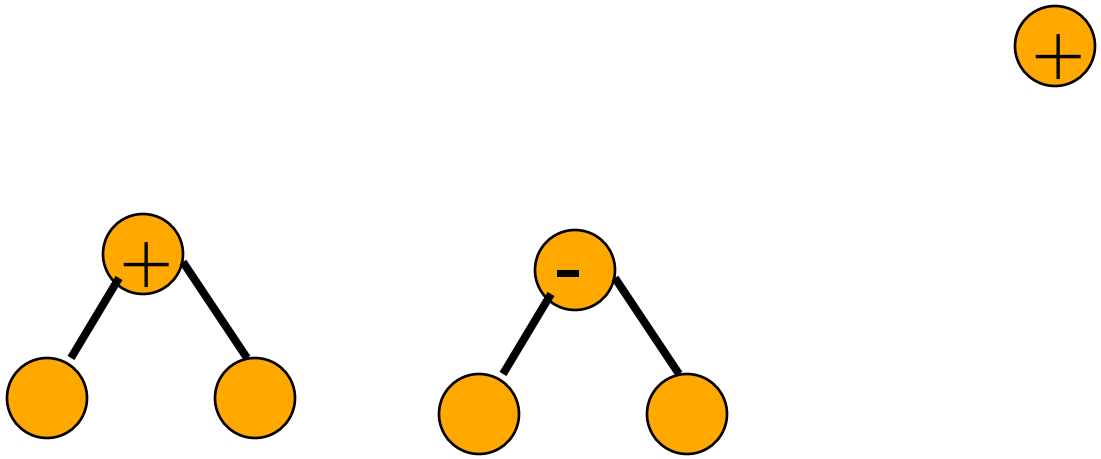
-

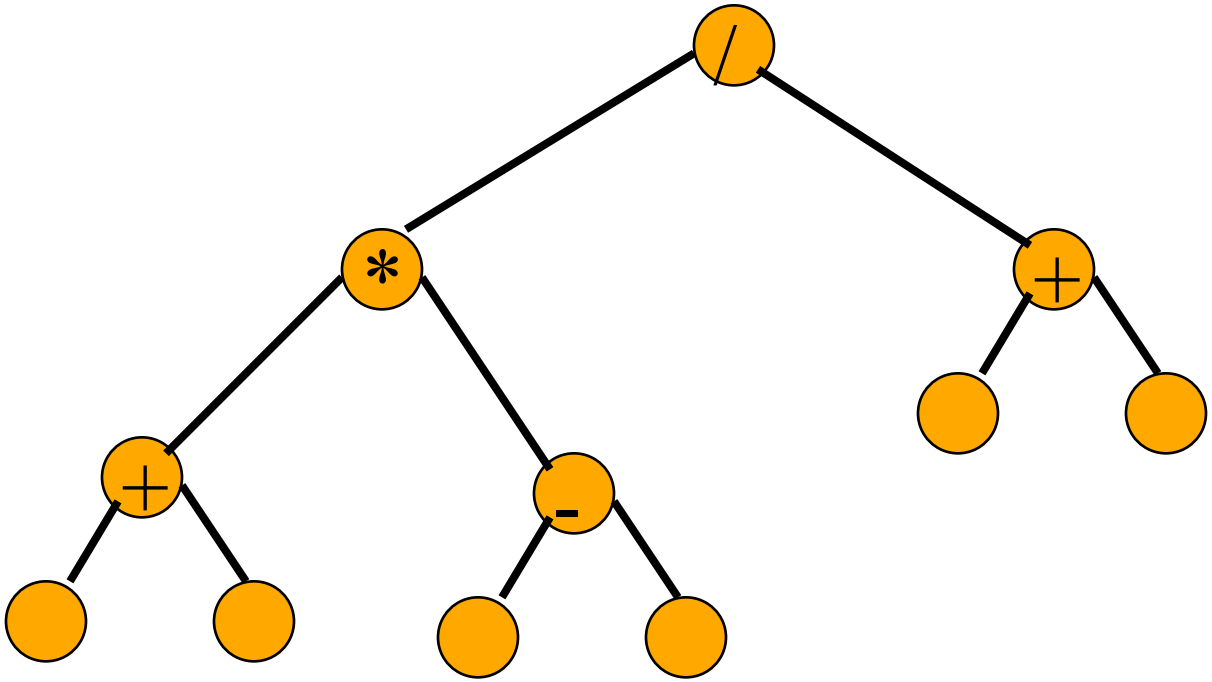




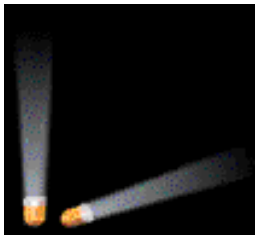
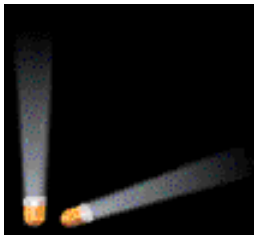
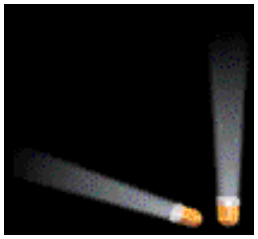
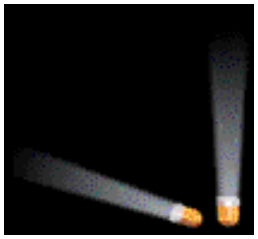
(+) * ( ) / (+)

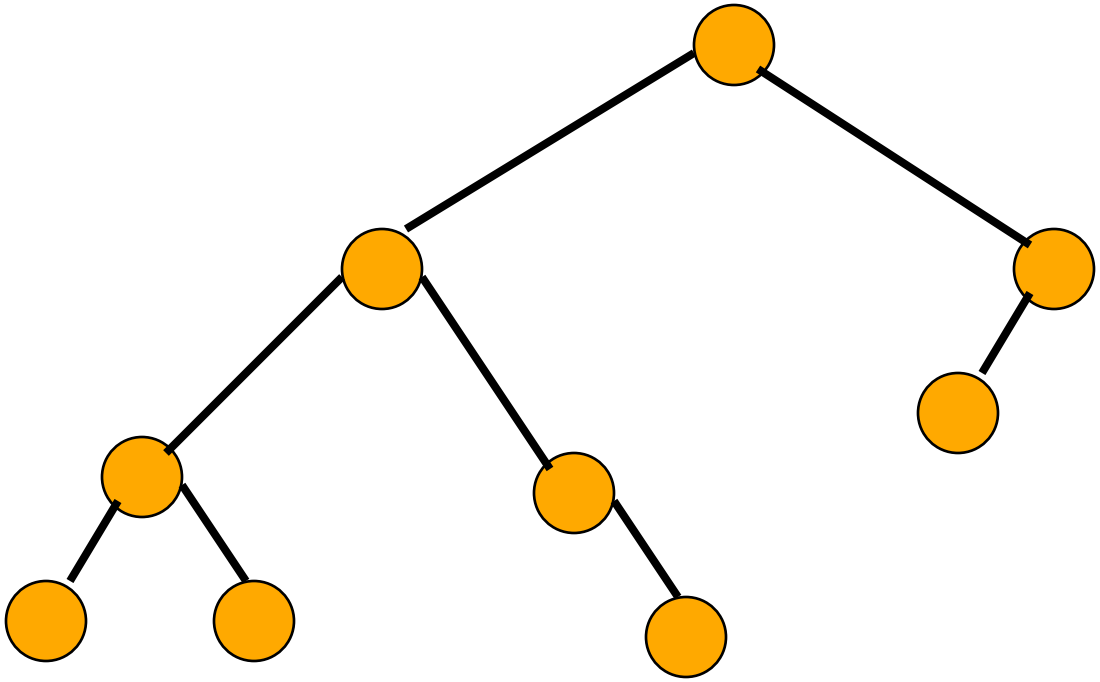
/



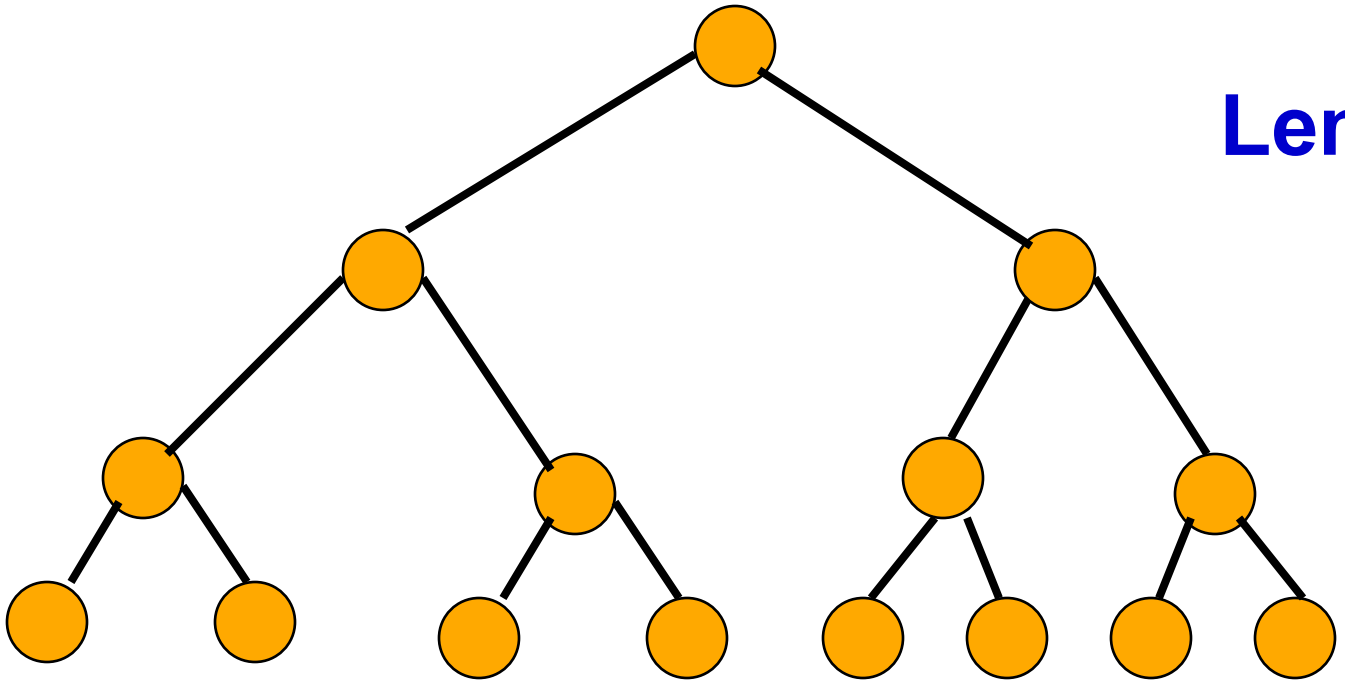


&



Lemma 5.2



$= 1 + 2 + 4 + 8 + \quad + 2^{-1}$

$= 2 - 1$

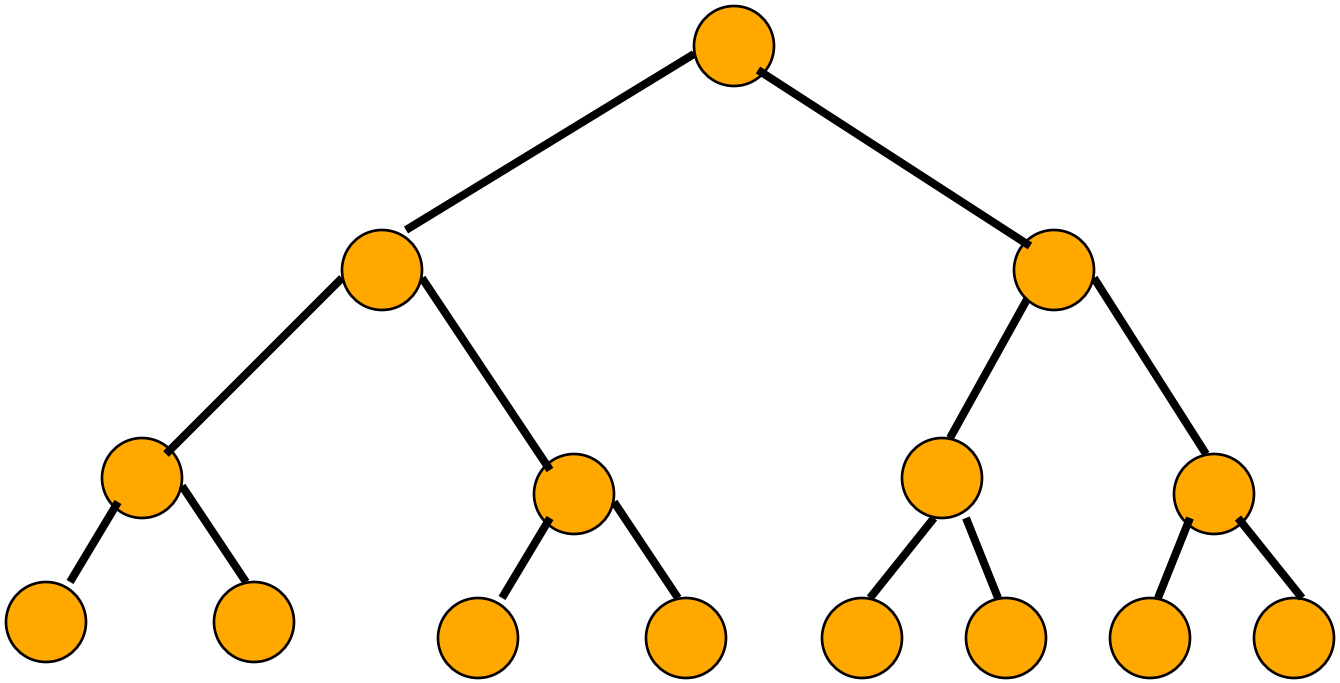
&

.

<= <= 2 1

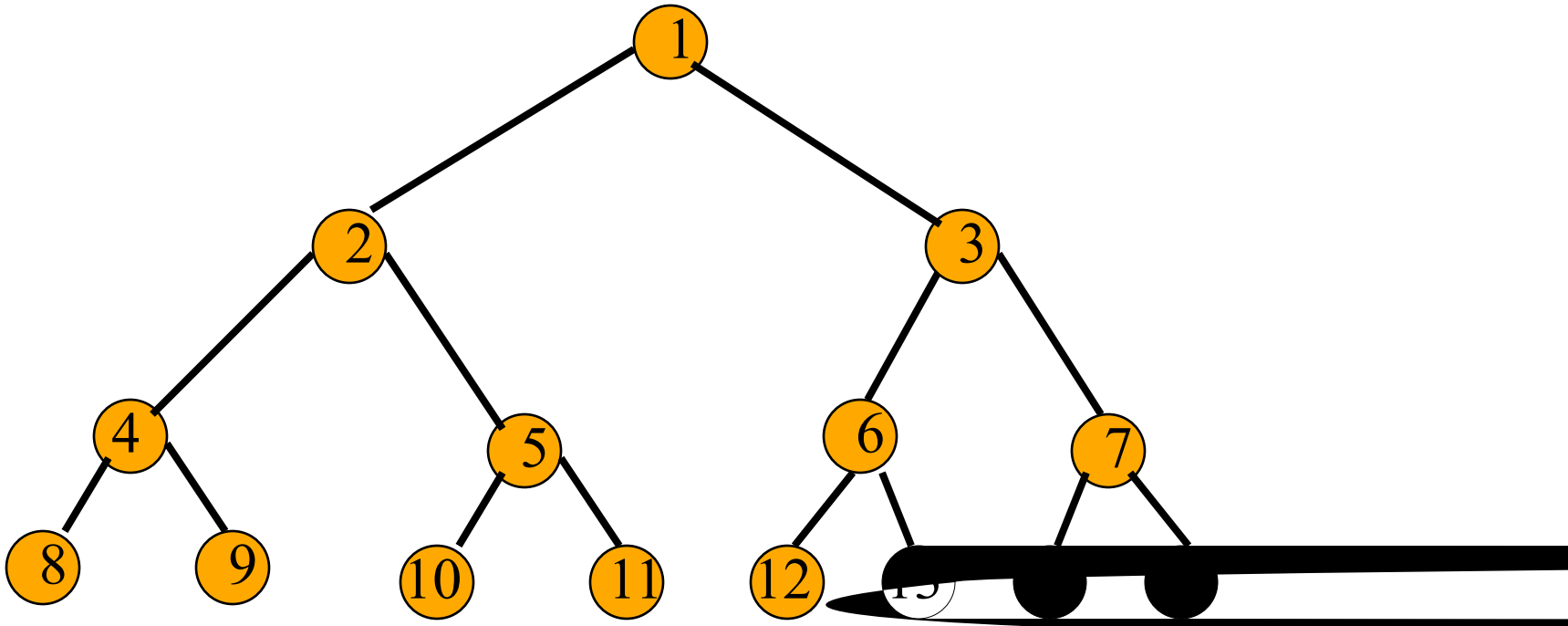
$_2$(+1) <= <=

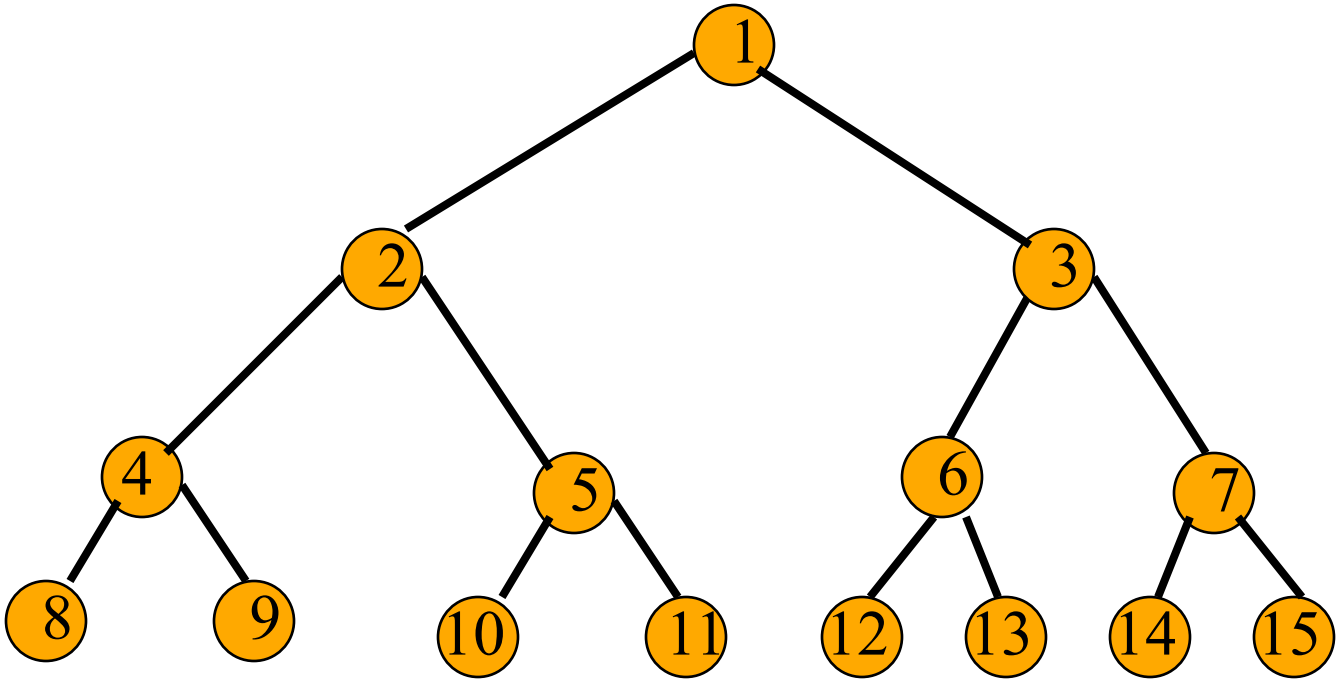
2 1



4

1 2 1.





/ 2, = 1.

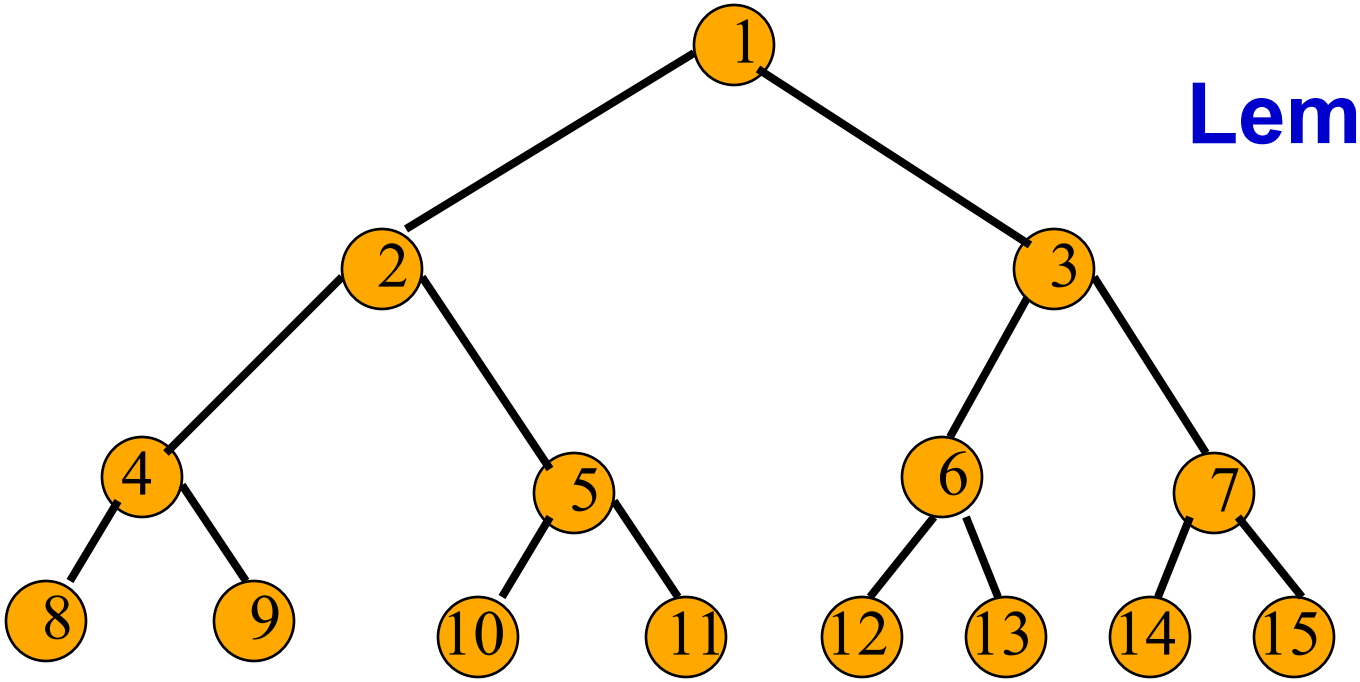
1

.

2 , 2 > ,

.

>



Lemma 5.4

2 +1, 2 +1

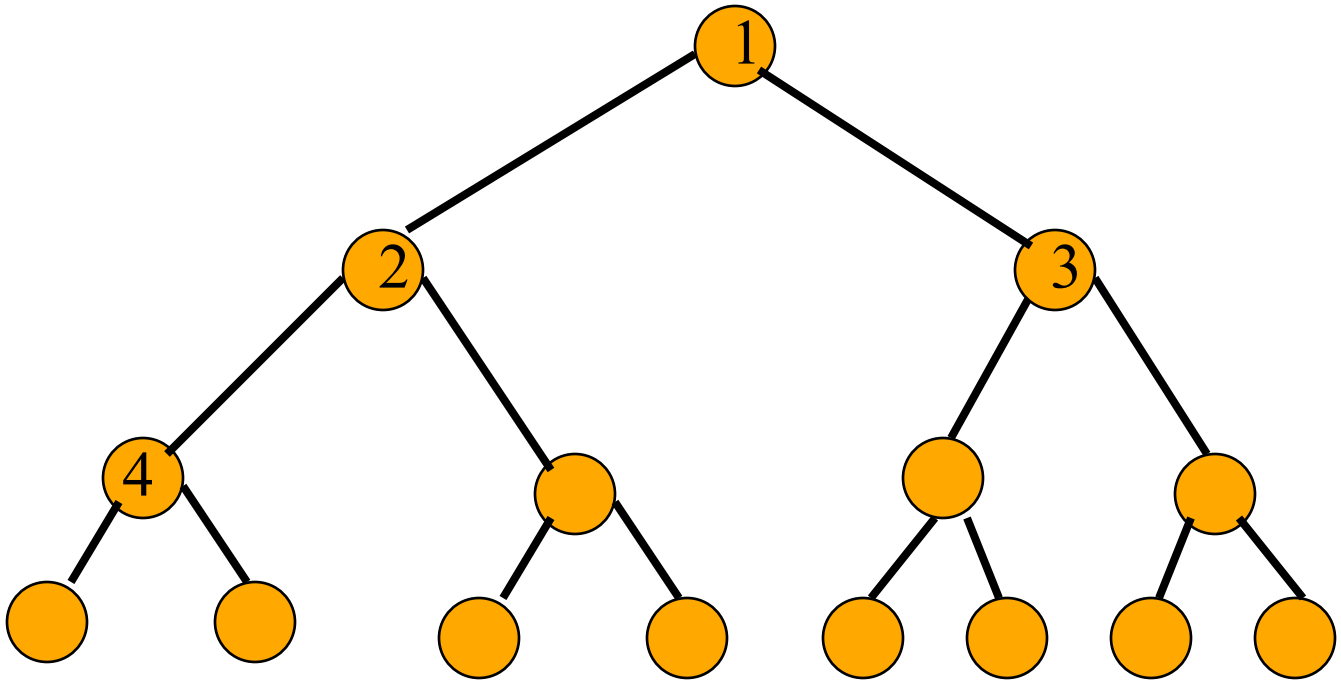
> ,

.

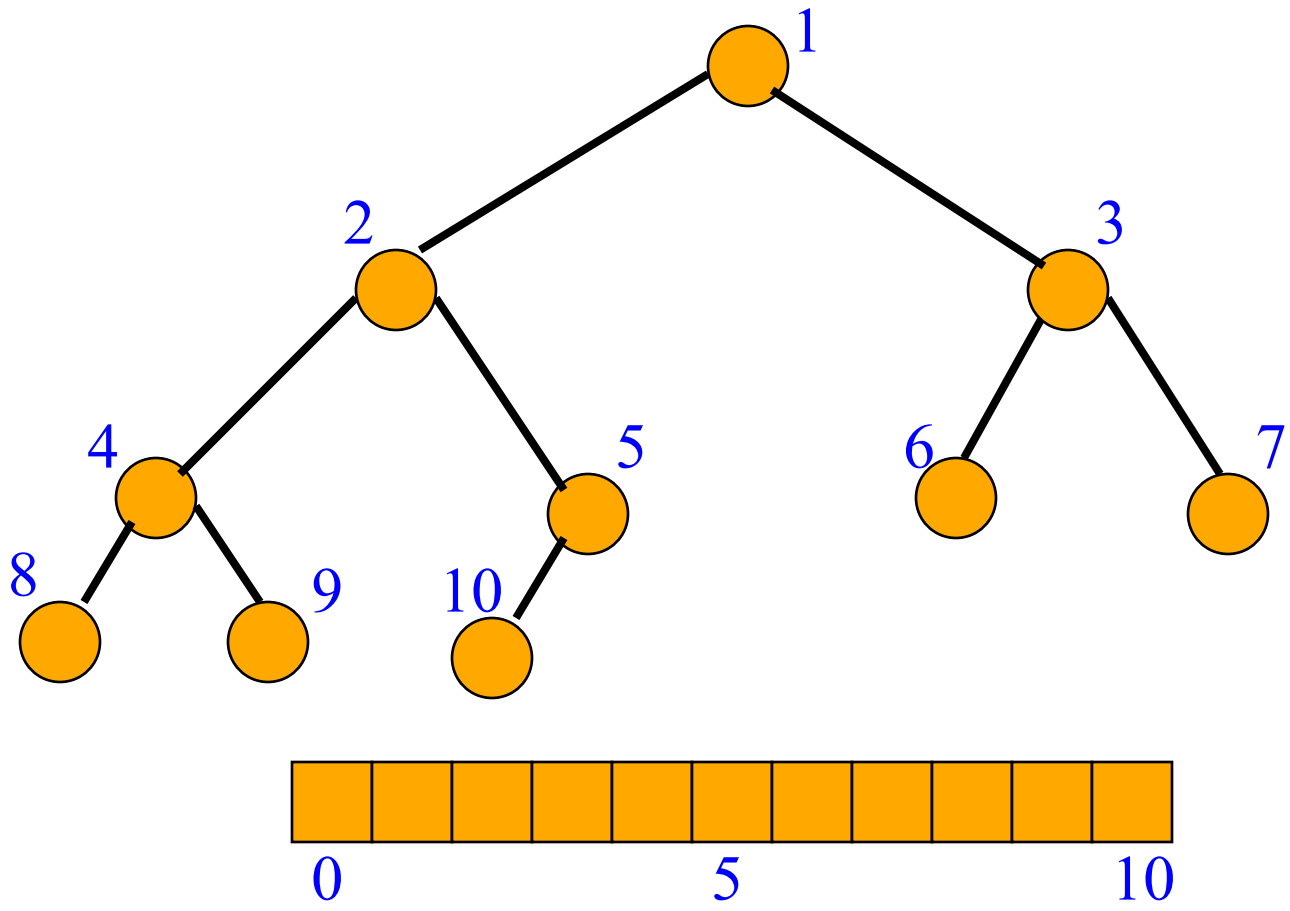
2 +1 > ,

.

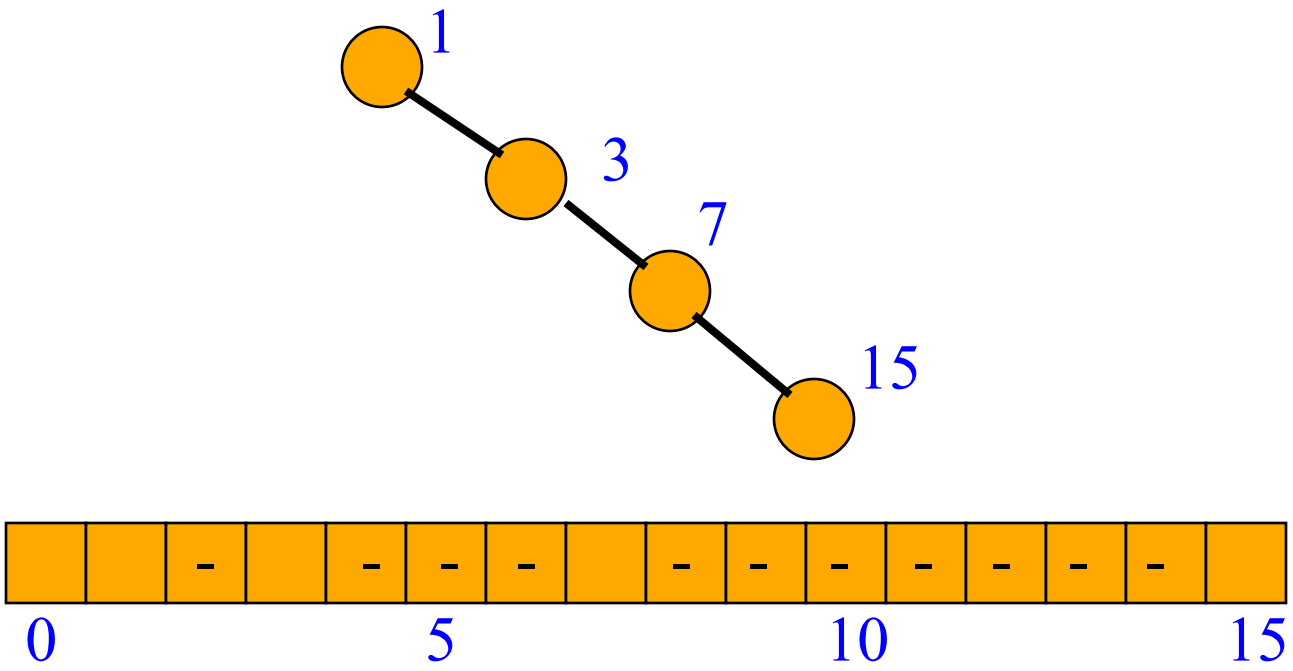
Definition: a binary tree with n nodes and depth k is **complete** iff its nodes corresponding to the nodes numbered from 1 to n in the full binary tree of depth k.



10 .


1
2
3
4
5
6
7
8
9
10
0
5
10

-



+1 2 .

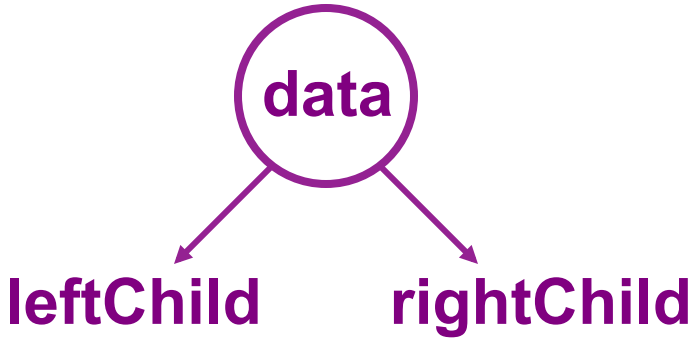
.

* ( ).

. .

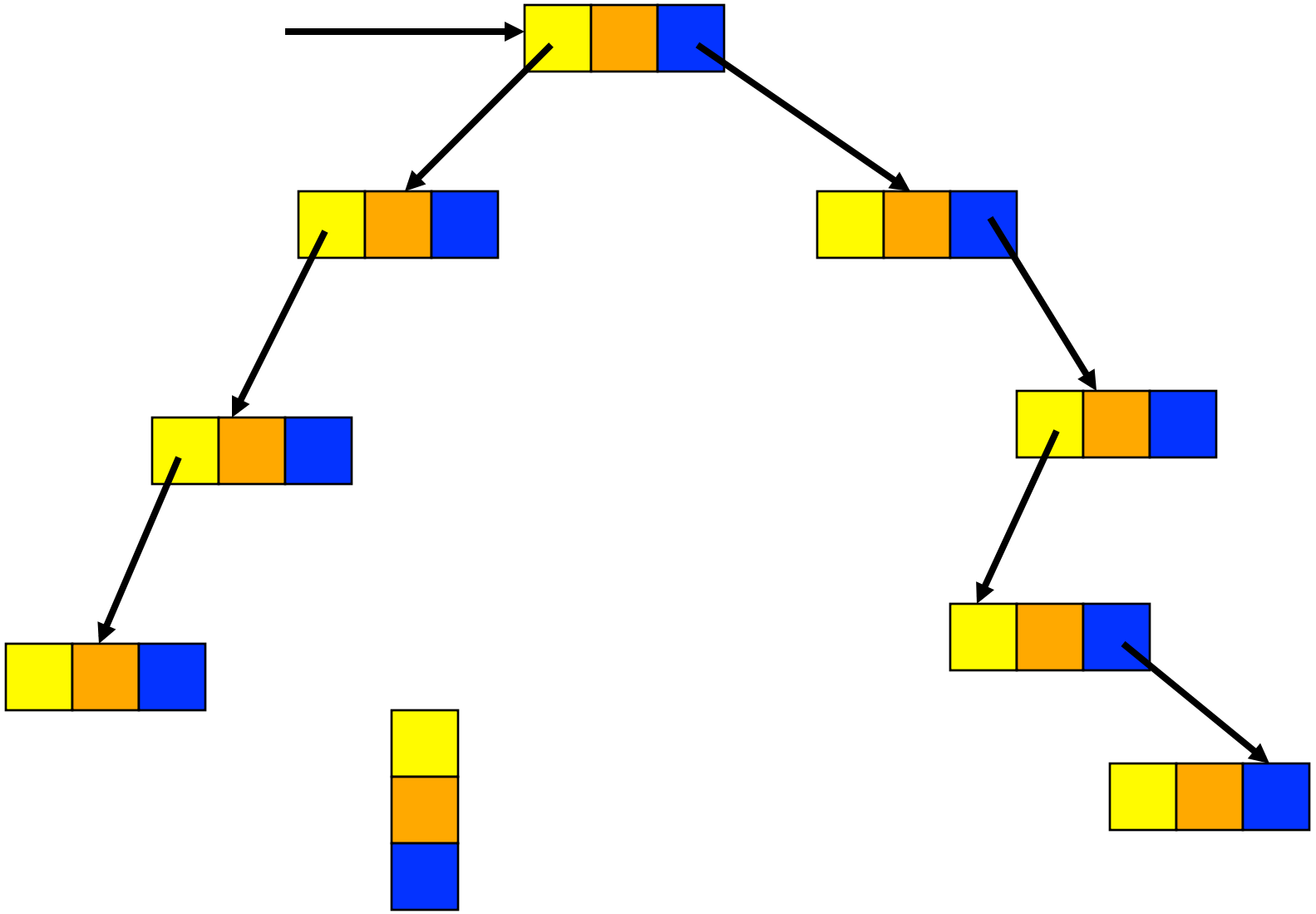
```
template <class T> class Tree;
class TreeNode {
friend class Tree<T>;
public:
    TreeNode (T& e, TreeNode<T>* left, TreeNode<T>* right)
        {data=e; leftChild=left; rightChild=right;}
private:
    T data;
    TreeNode<Y>* leftChild;
    TreeNode<Y>* rightChild;
};
```

leftChild	data	rightChild

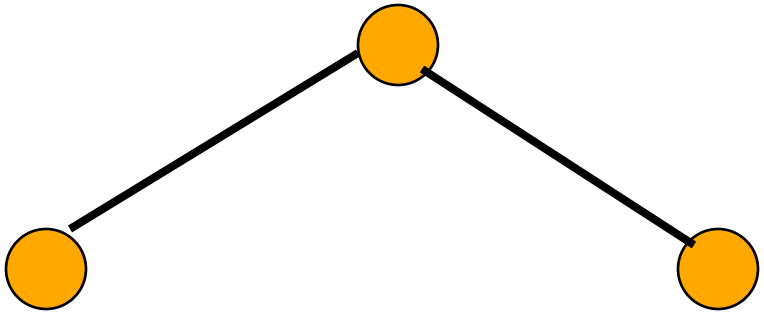



```
template <class T>
class Tree {
public:
    // Tree operations
…
private:
    TreeNode<T>* root;
};
```

If necessary, a 4th field, parent, may be included in the node.



(=)



< >

< >:: ()

// .

( )

< >

< >:: (< >*)

// .

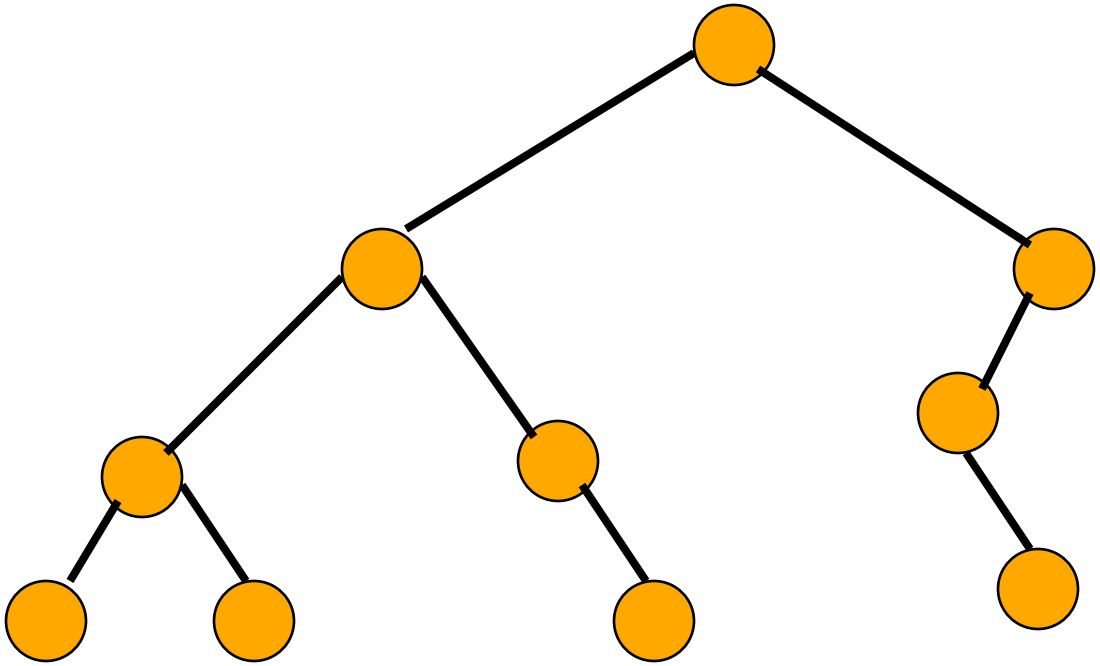
( )

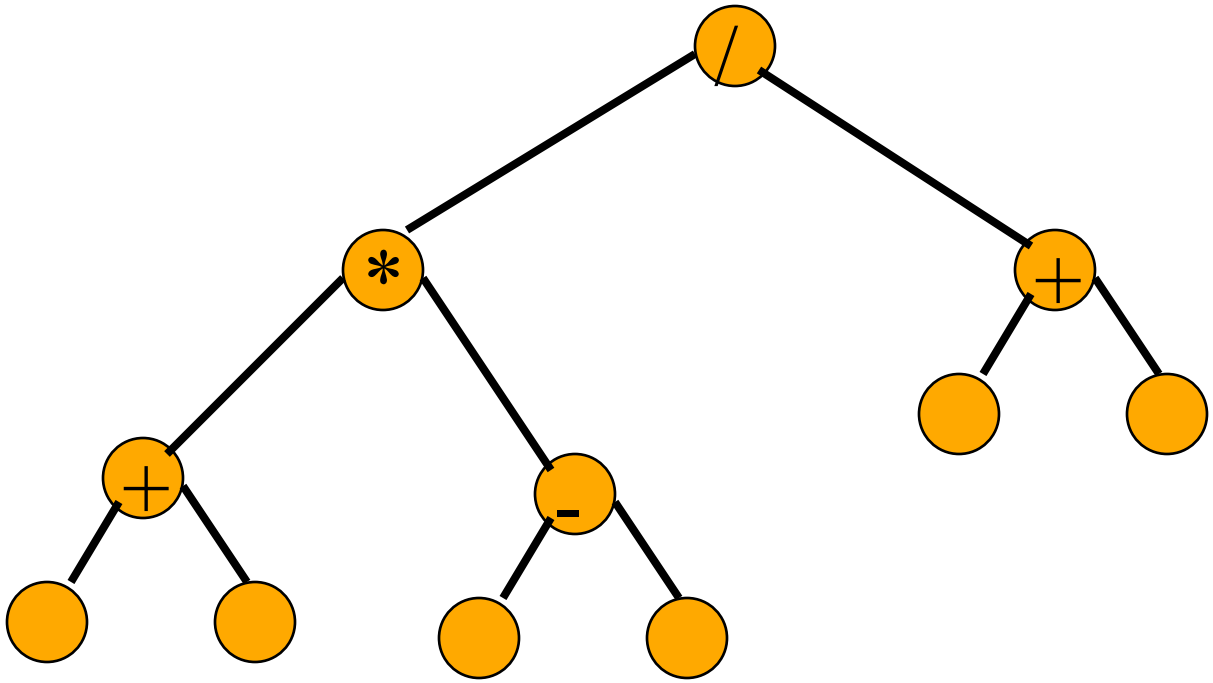
( )

(→)

(→)

(=)

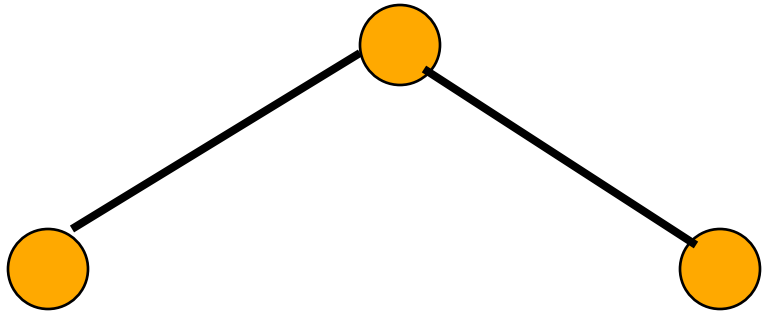




/ * + - +

!

(=)



< >

< >:: ()

//

( )

< >

< >:: (< >*)

//

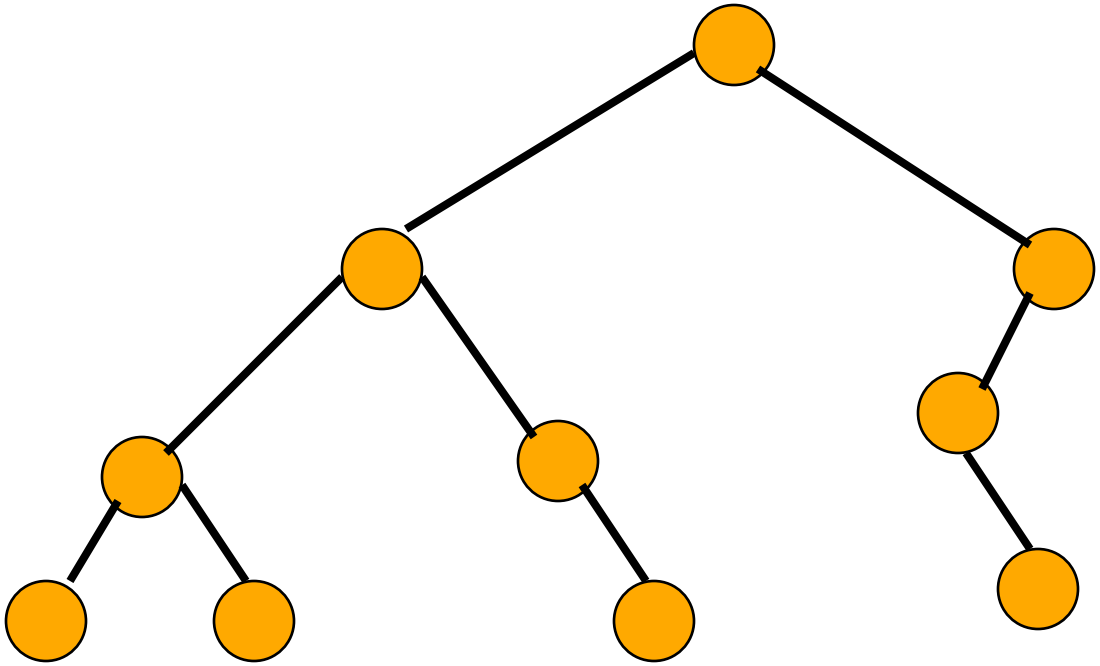
( )

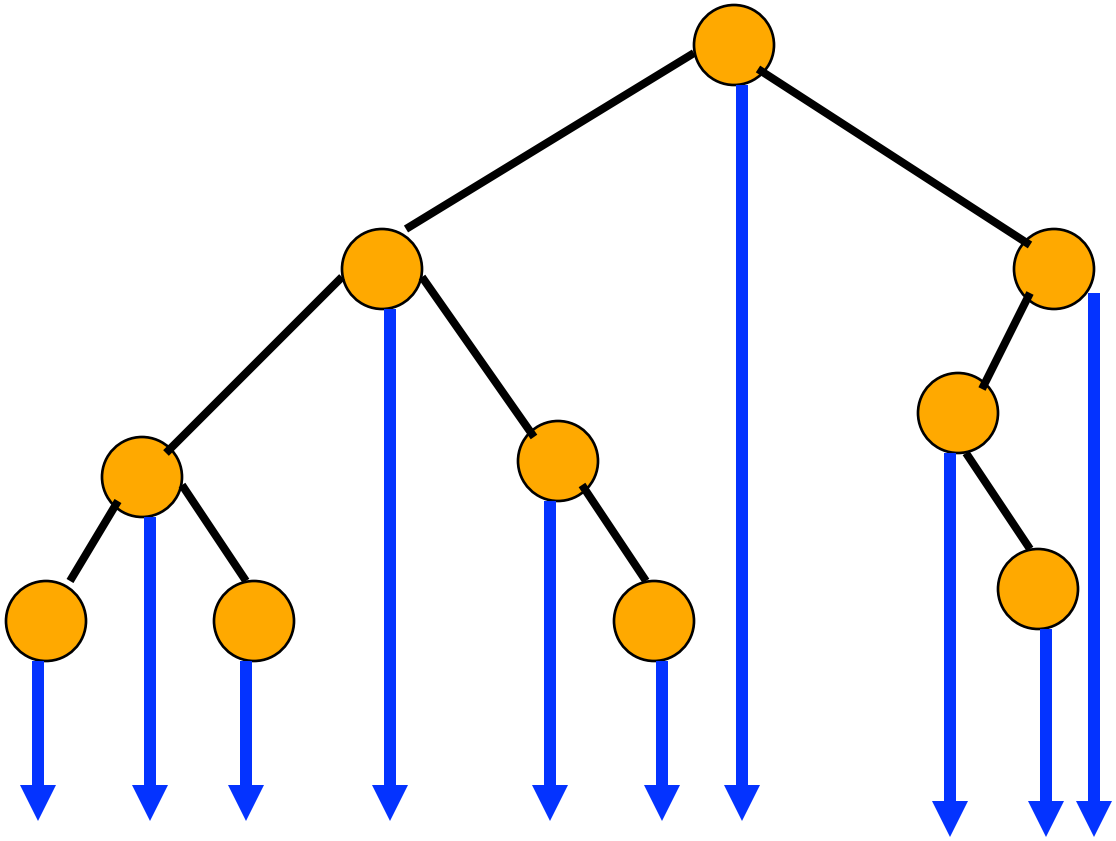
(→)

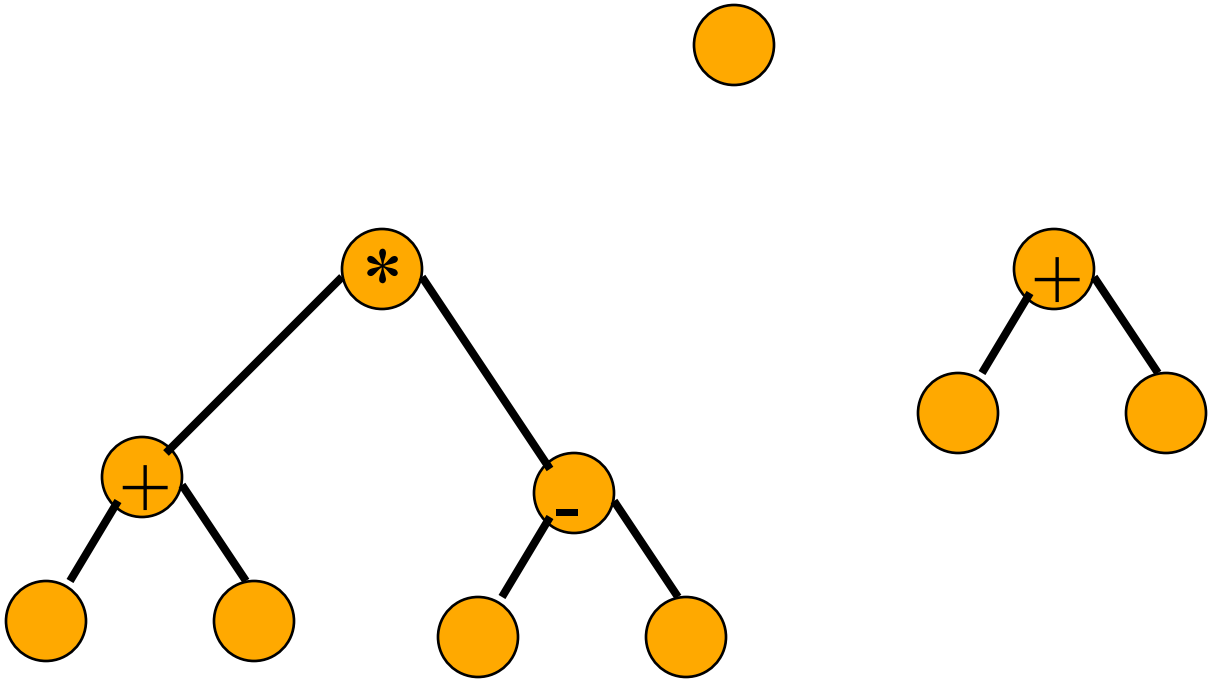
( )

(→)

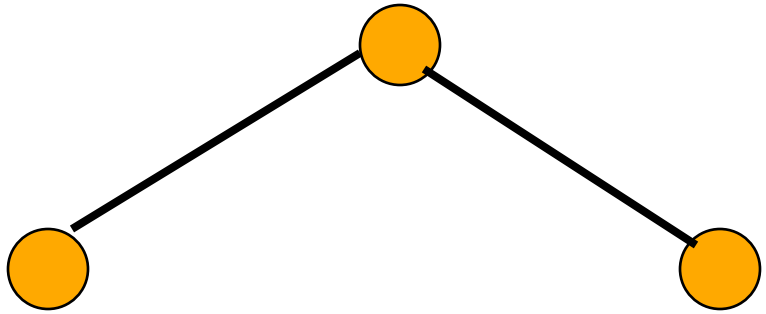
(=)







(=)



< >

< >:: ()

// .

( )

< >

< >:: (< >*)

// .

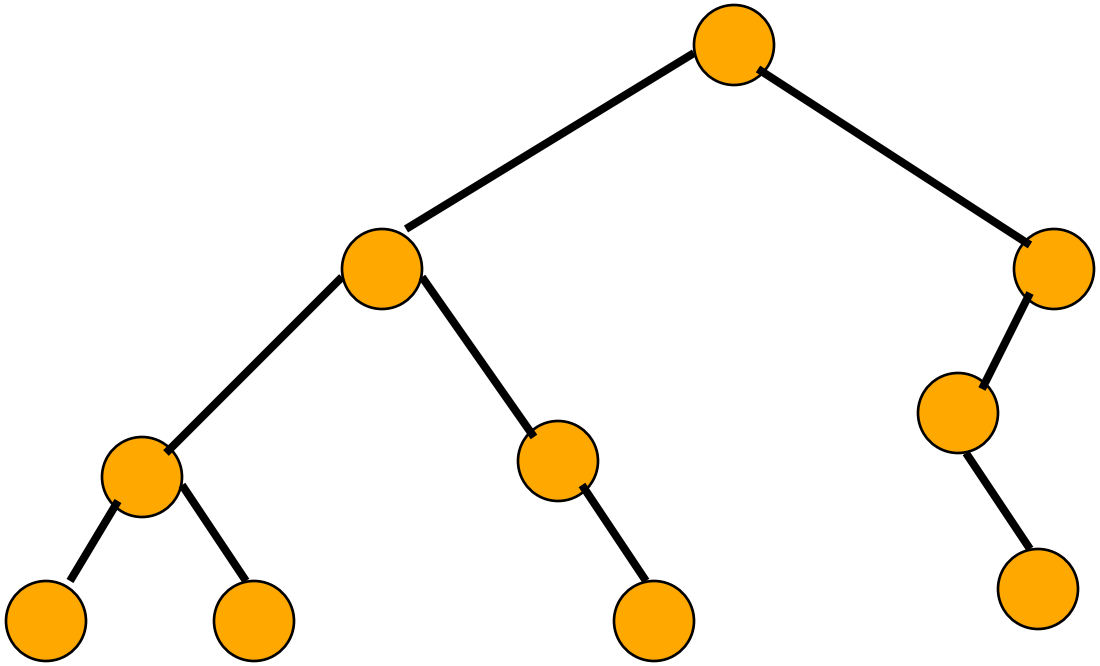
( )

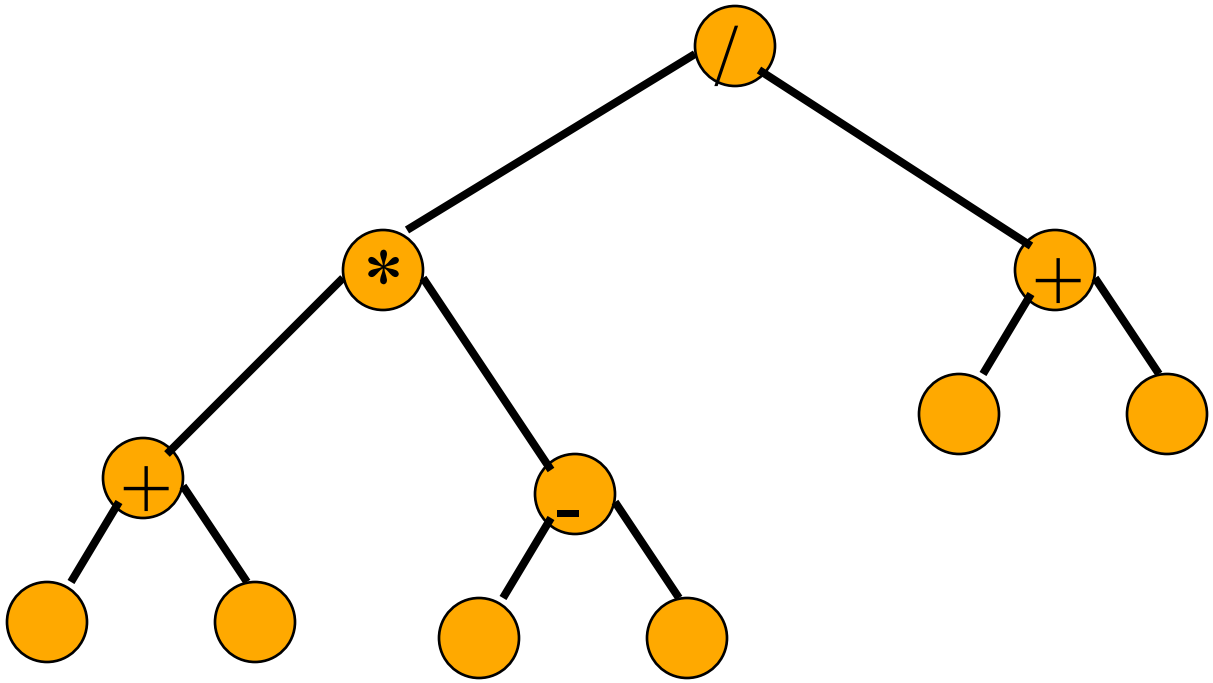
(→)

(→)

( )

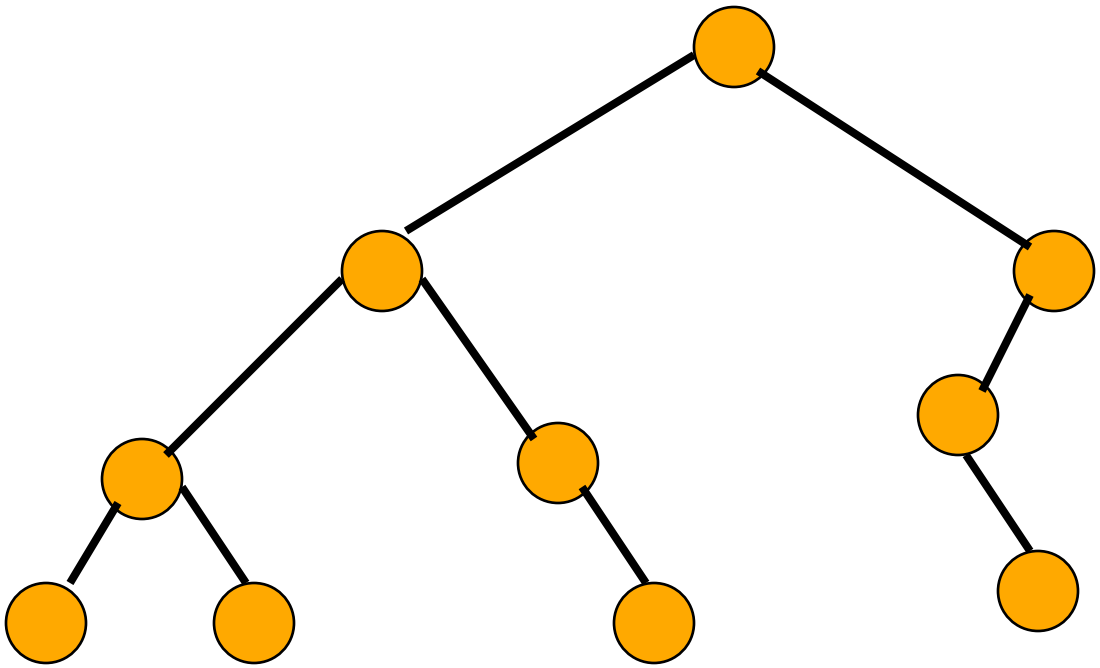
(=)





+ - * + /

!



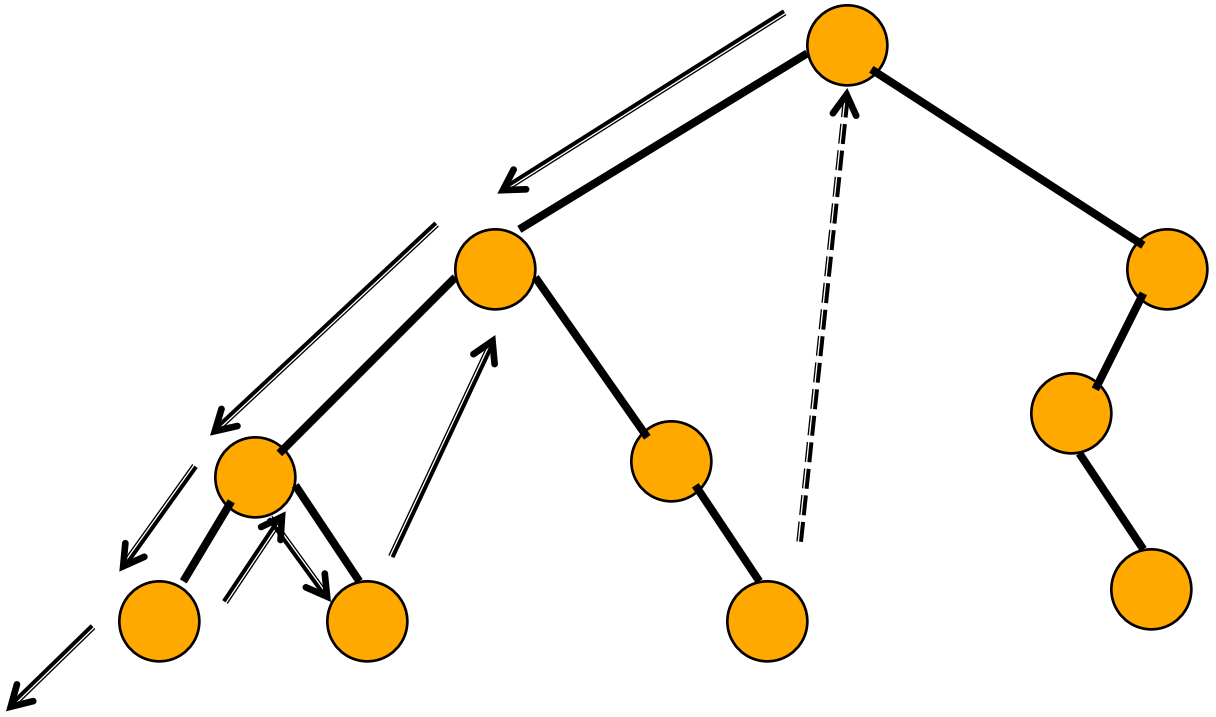
(*)

- (==) 0;
-
-

= (->);
= (->);
+ + 1;

-

Iterative Inorder Traversal



```
1 template <class T>
2 void Tree<T>::NonrecInorder()
3 { // Nonrecursive inorder traversal using a stack
4   Stack<TreeNode<T>*> s; // declare and initialize a stack
5   TreeNode<T>* currentNode=root;
6   while (1) {
7     while (currentNode) { // move down leftChild
8       s.Push(currentNode); // add to stack
9       currentNode=currentNode→leftChild;
10    }
11    If (s.IsEmpty()) return;
12    currentNode=s.Top();
13    s.Pop(); // delete from stack
14    Visit(currentNode);
15    currentNode=currentNode→rightChild;
16  }
17}
```

//

() =

* ( )

< < >*>

< >*

:: ()

( )

. ( )

= →

(. ()) 0

= . ()

& = →

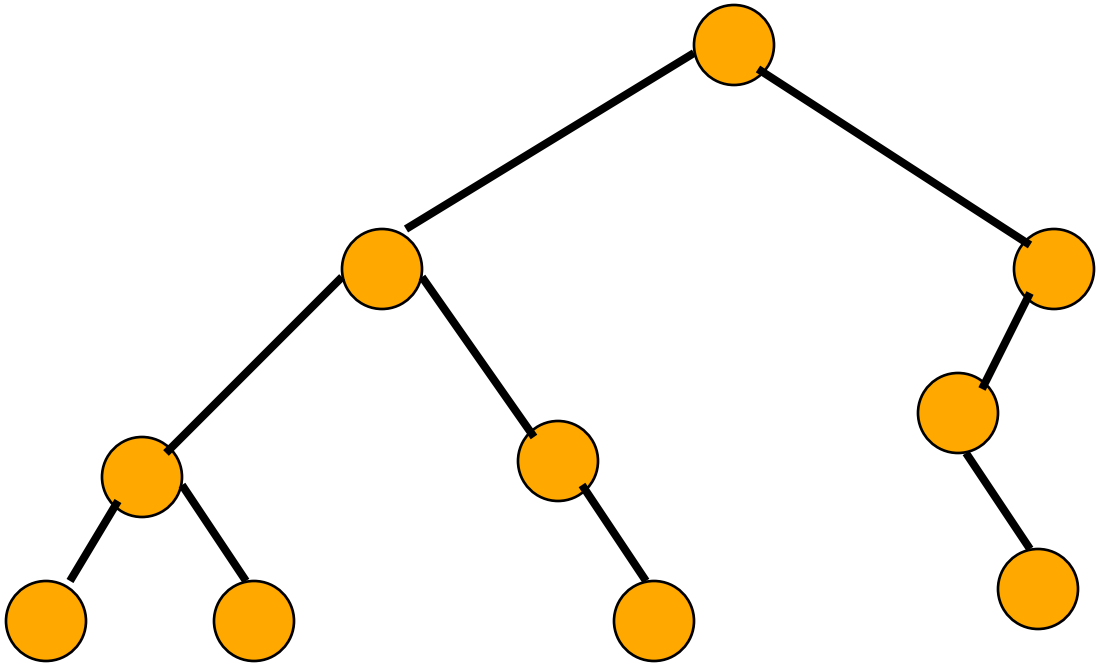
= →

&

}

-

(=)



.

(!=)

;

;

//

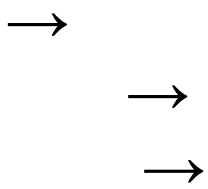
;

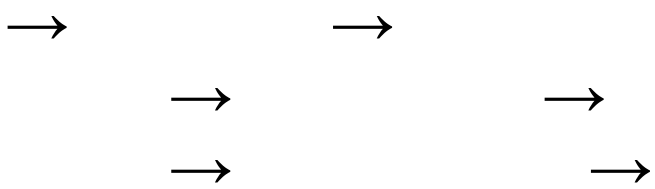
. ( );

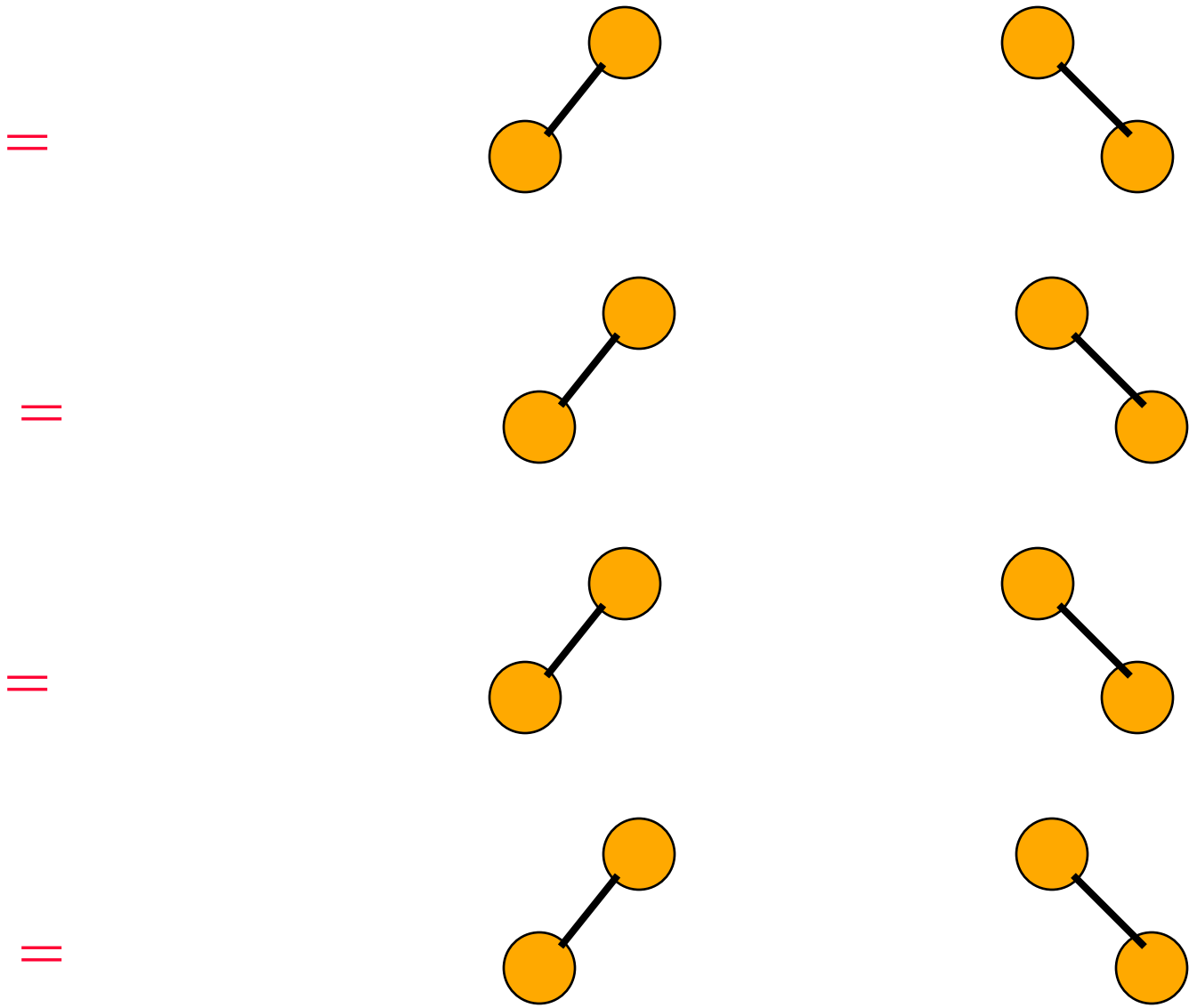
(! . ())

- * = . ();
- ( );
- (->) . (->);

(->) . (->);





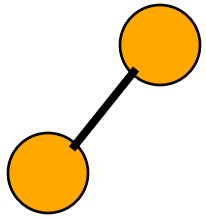


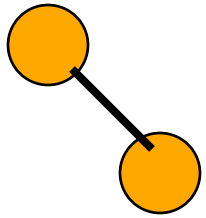
،

.

=

=

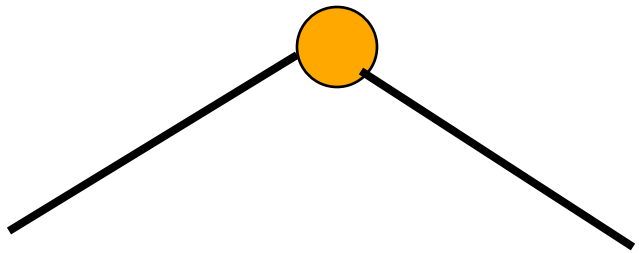


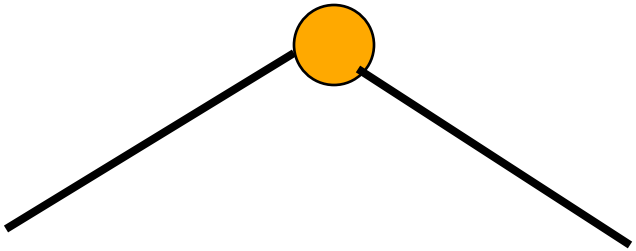


.

( ).

( ).



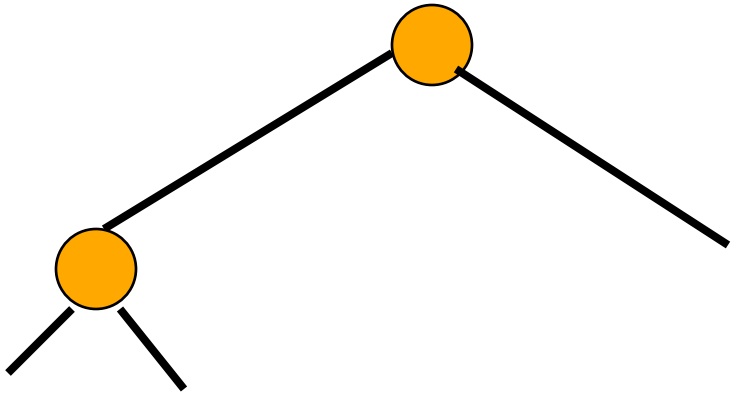


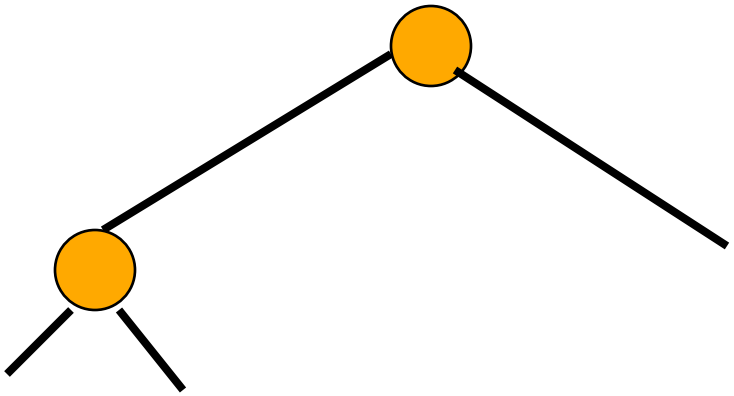
=

;

;

.

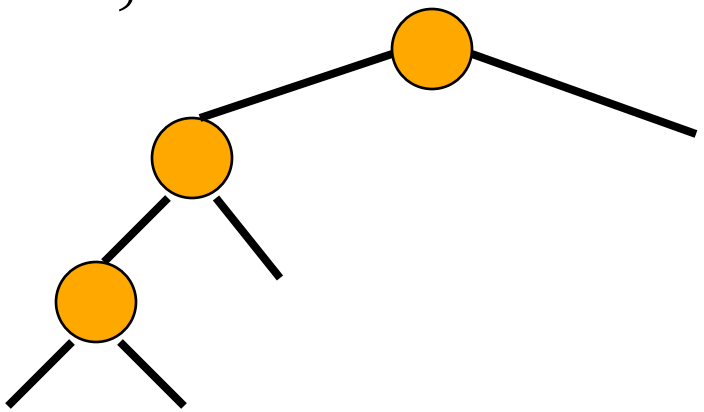


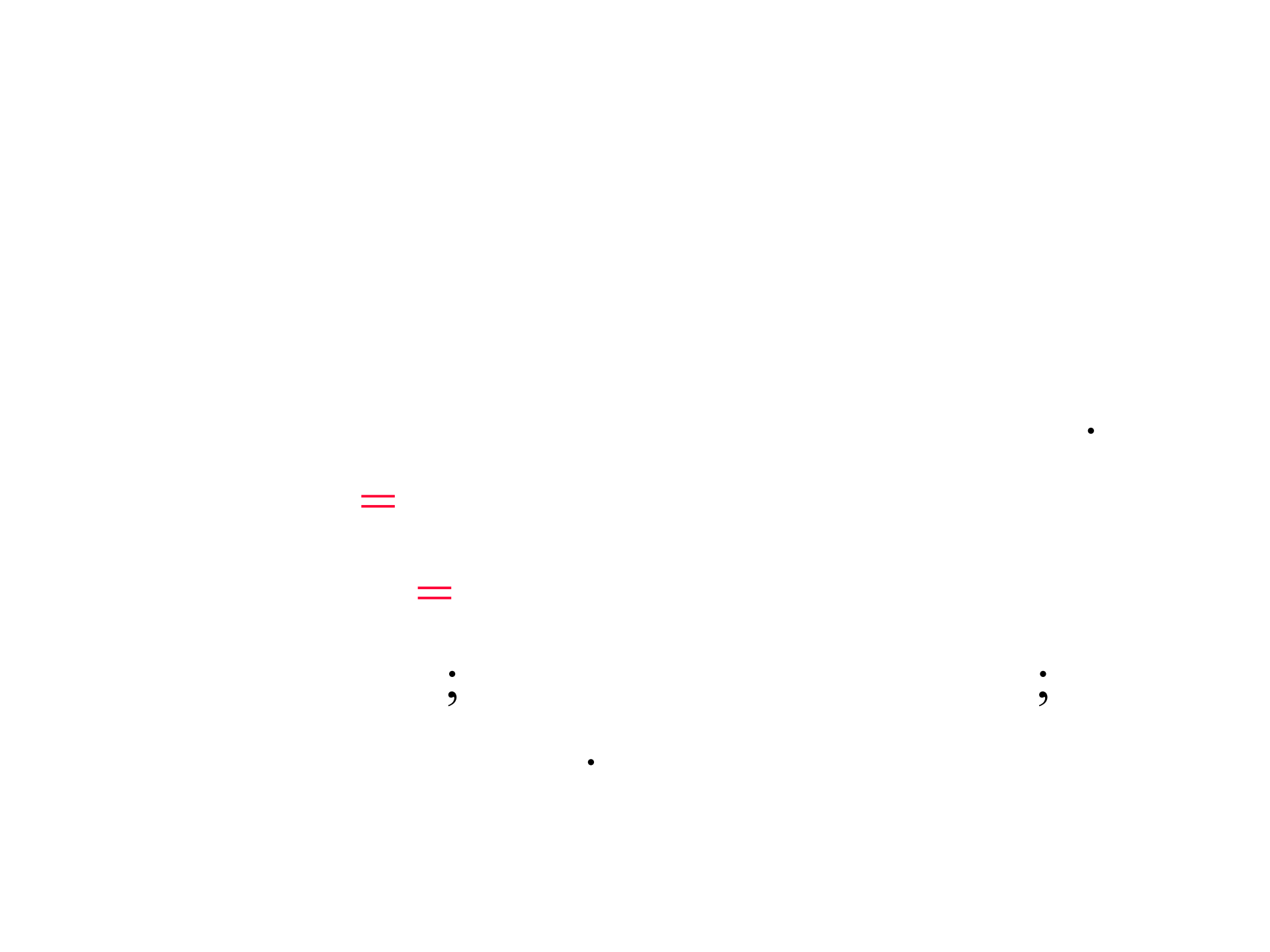


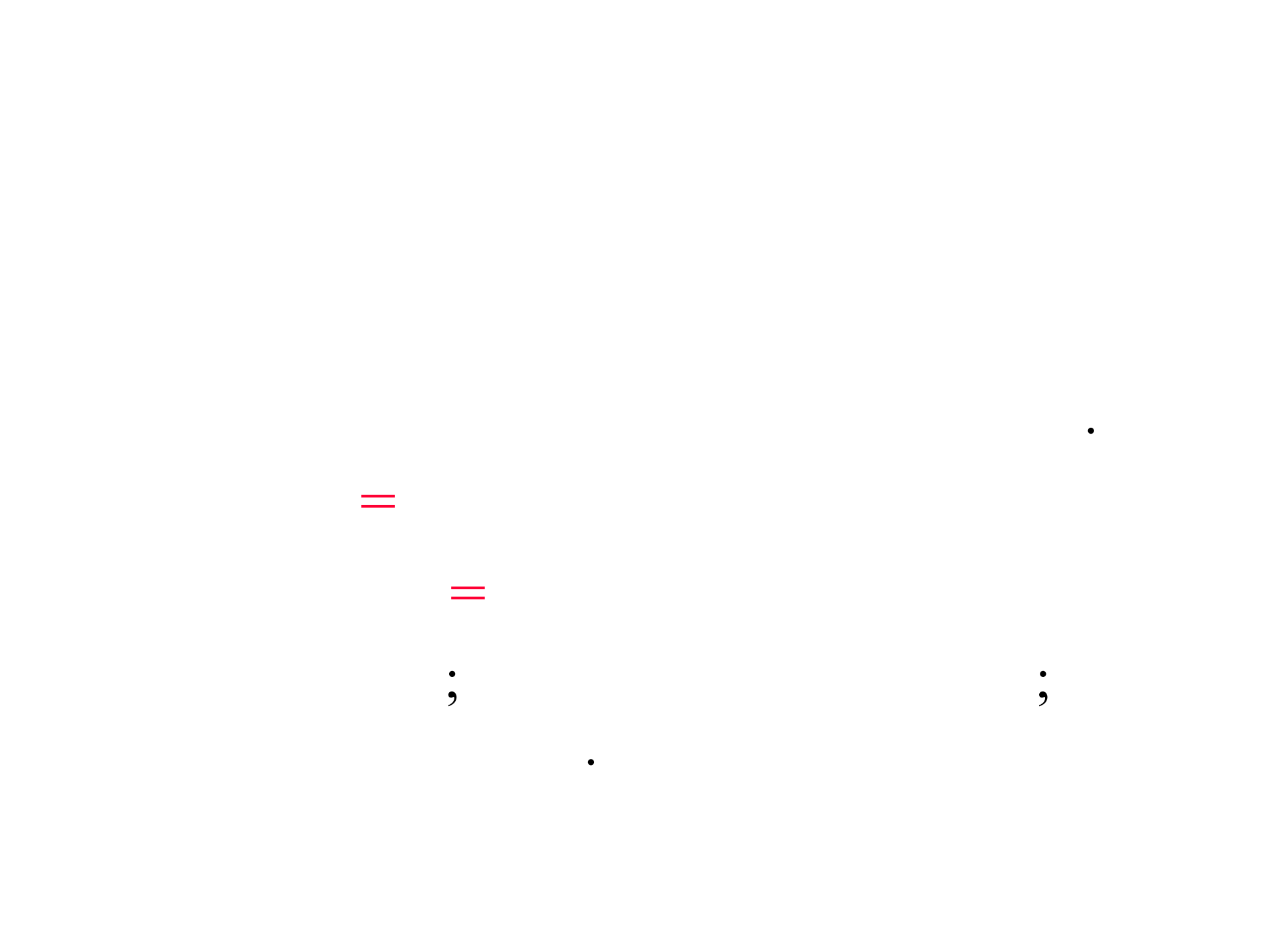
=

;

;







Exercises:

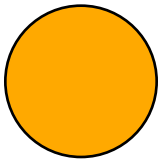
1. , = , = , .

2. 3, = , = .

=

,

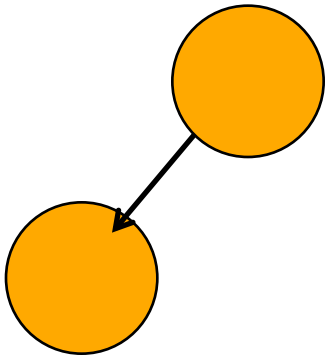
=



=

,

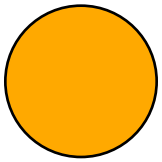
=



=

,

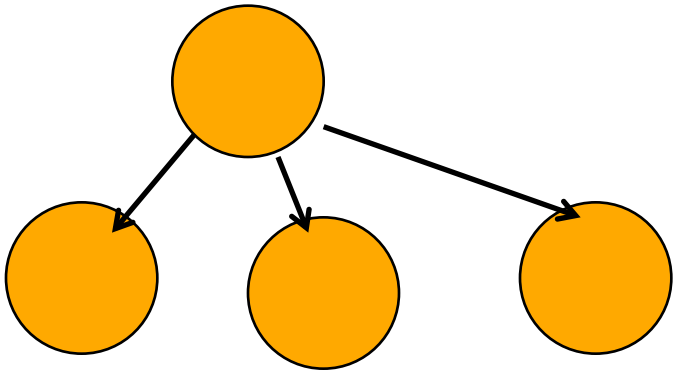
=



=

,

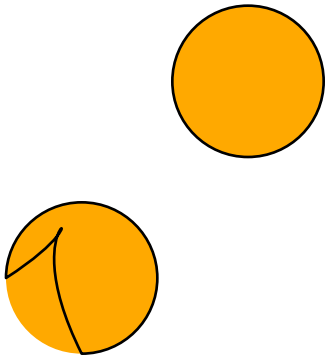
=



=

,

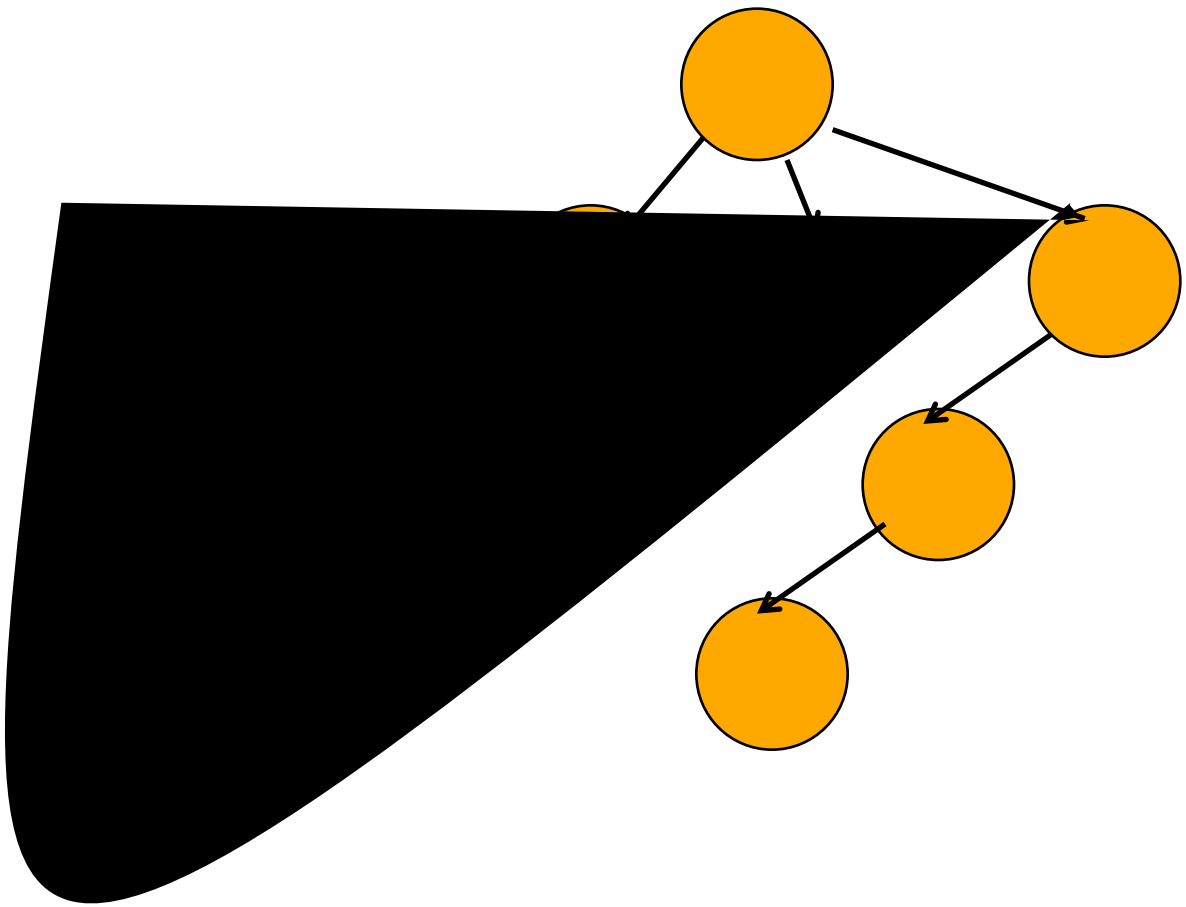
=



=

,

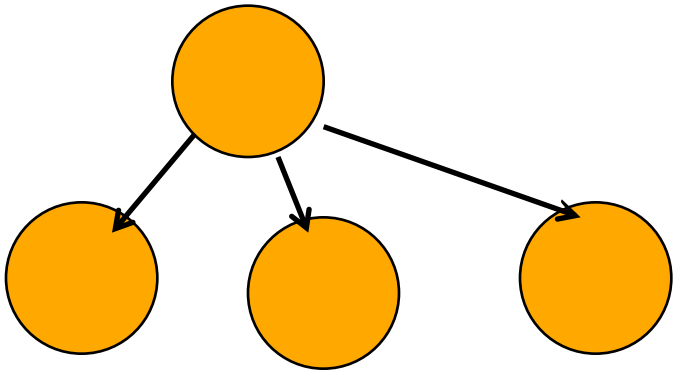
=



=

,

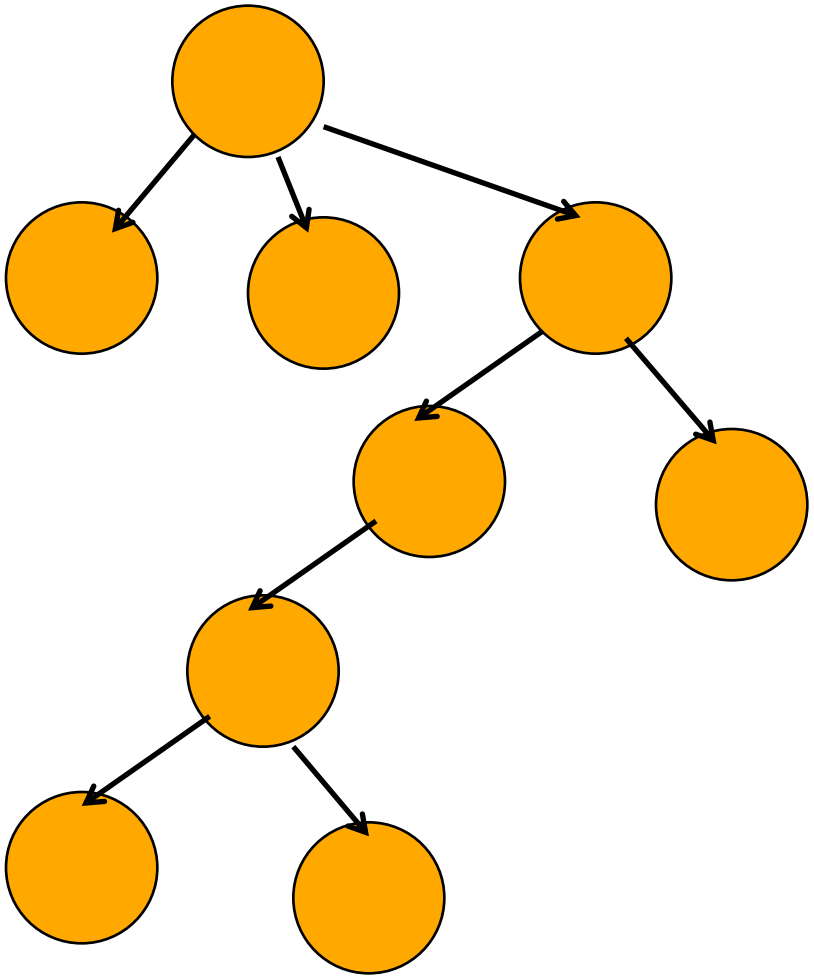
=



=

,

=



:

- ( )
- (,)
- ( )

:

- ()
- ( ) ( )
- ( ) ( )

-

< , >

() = 0

//

< , >* (&) = 0

// ;

// 0

(< , >&) = 0

// ;

//

(&) = 0

//

$$< \quad , \quad >$$

.

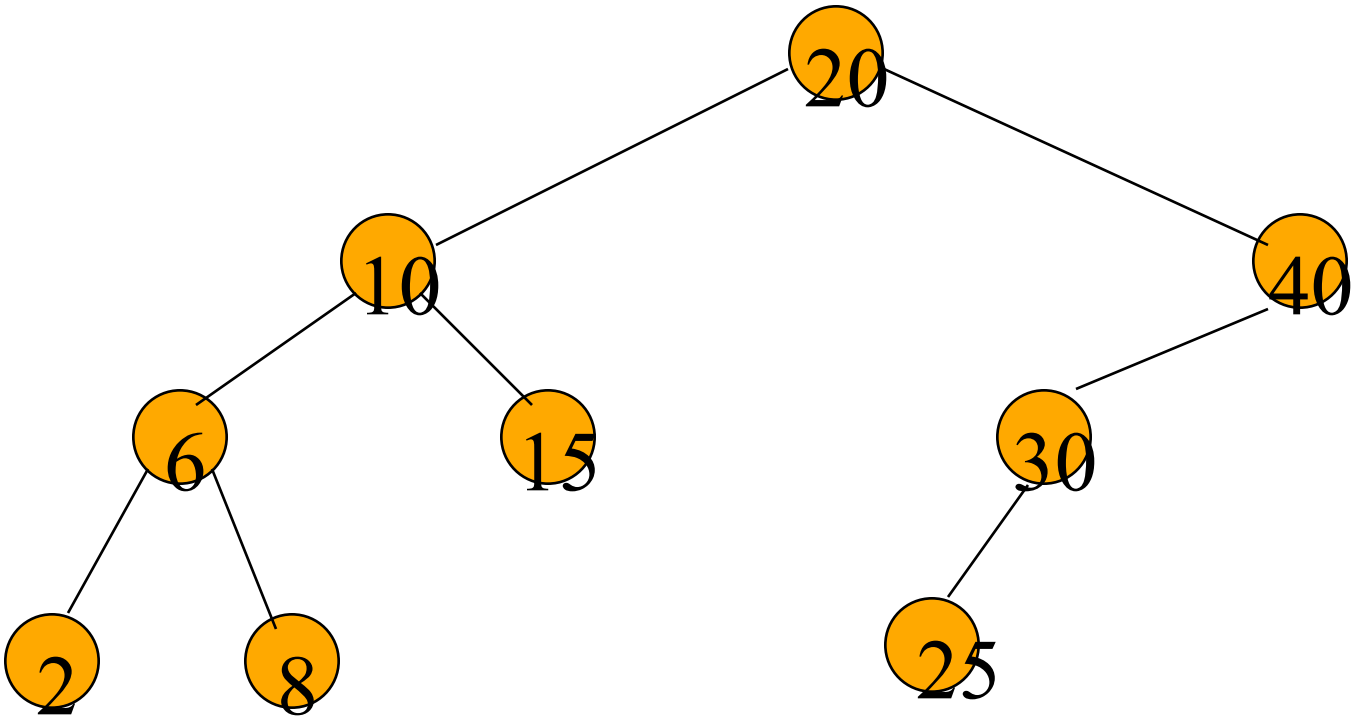
(,) .

,

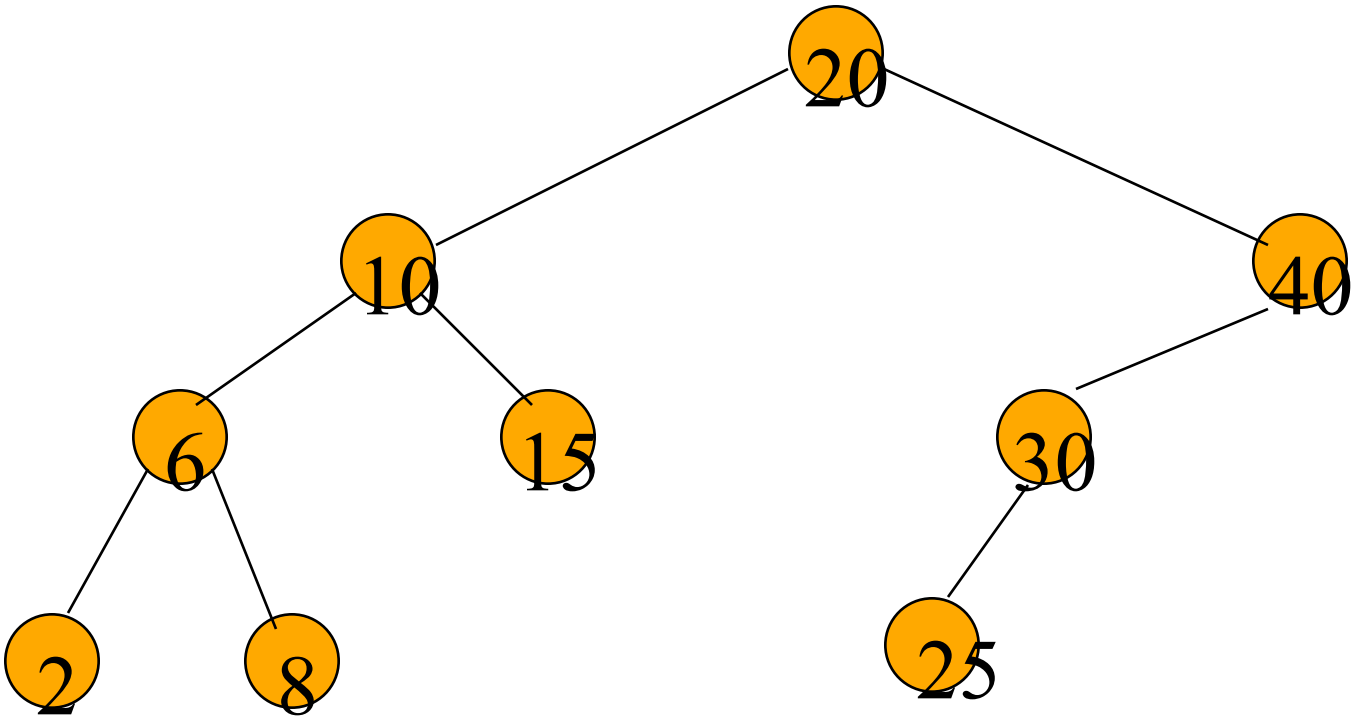
.

,

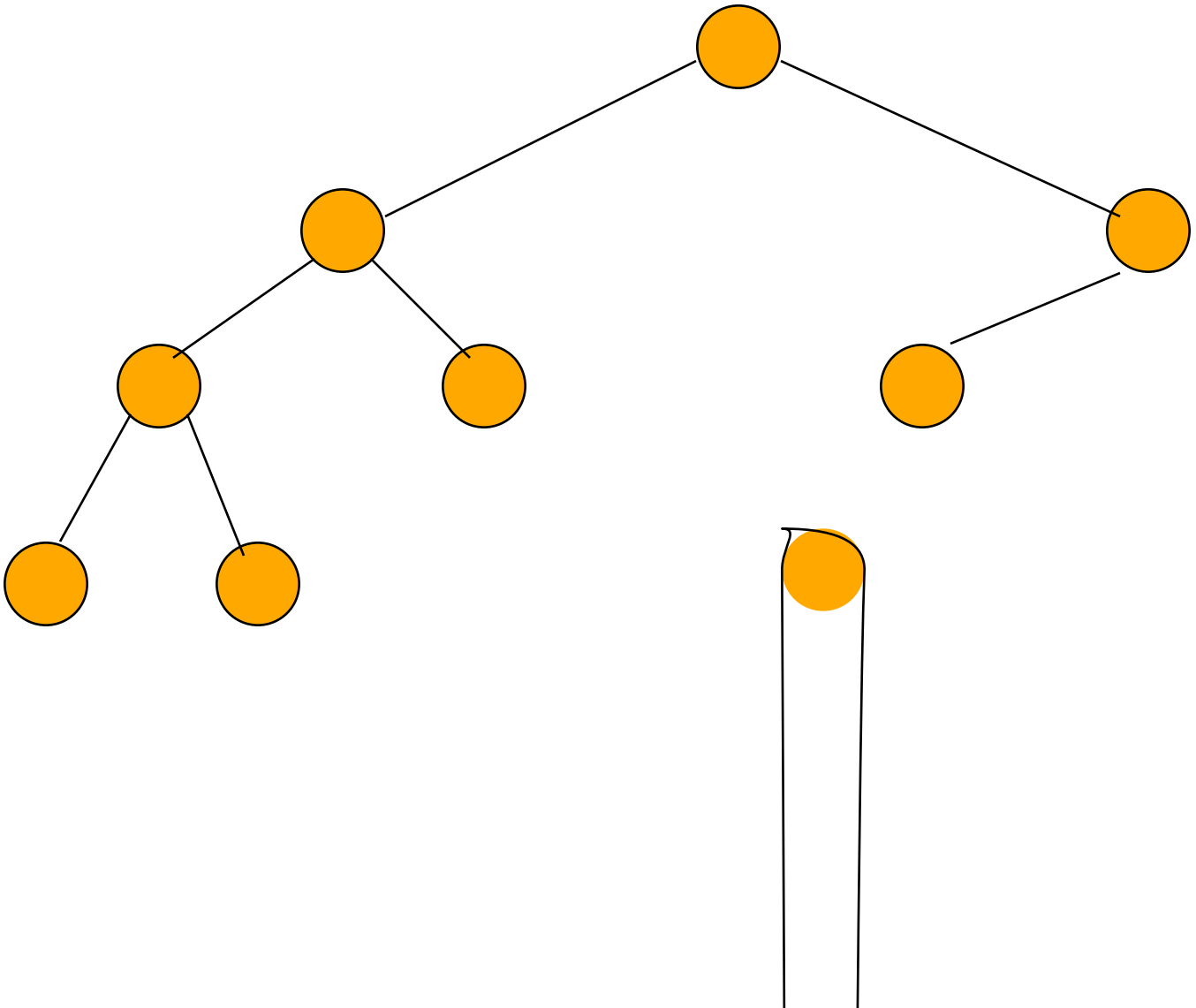
.


20
10
40
6
15
30
2
8
25

()


20
10
40
6
15
30
2
8
25

. ( ) .



()

-
-
-

()

< , > //
< , >* < , >:: (&)
// *
.
(,)

< , > //
< , >* < , >:: (< < , >>* ,
&)

(!) 0
(< → .) (→ ,)
(> → .) (→ ,)
& →

< , > //

< , >* < , >:: (&)

< < , >>* =

( )

(< → .)

= →

(> → .)

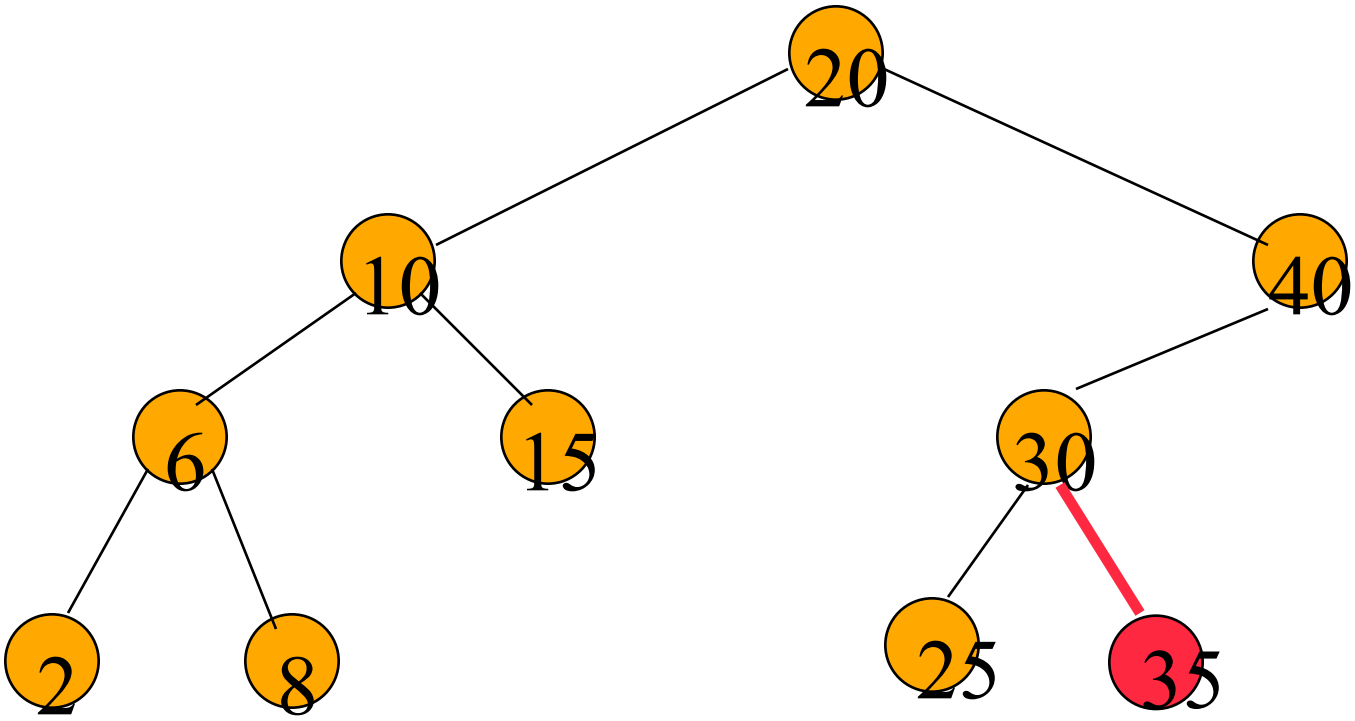
= →

& →

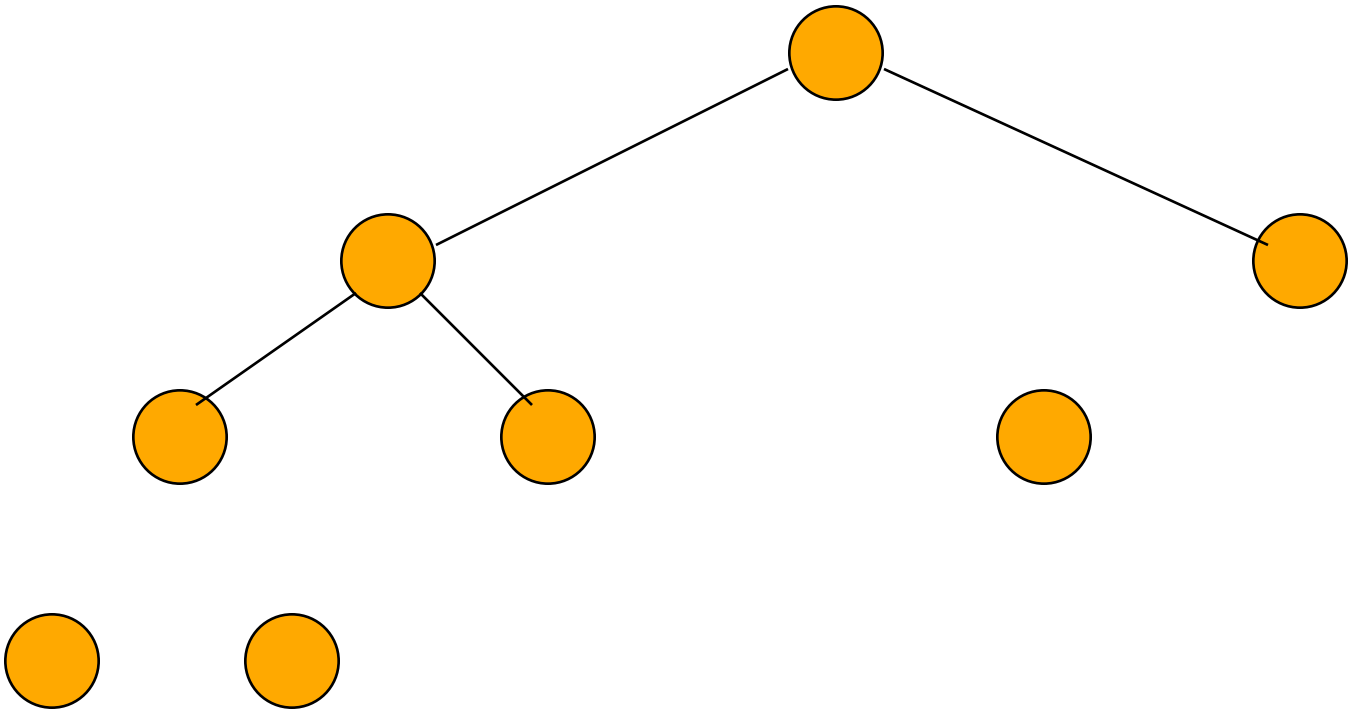
//

0

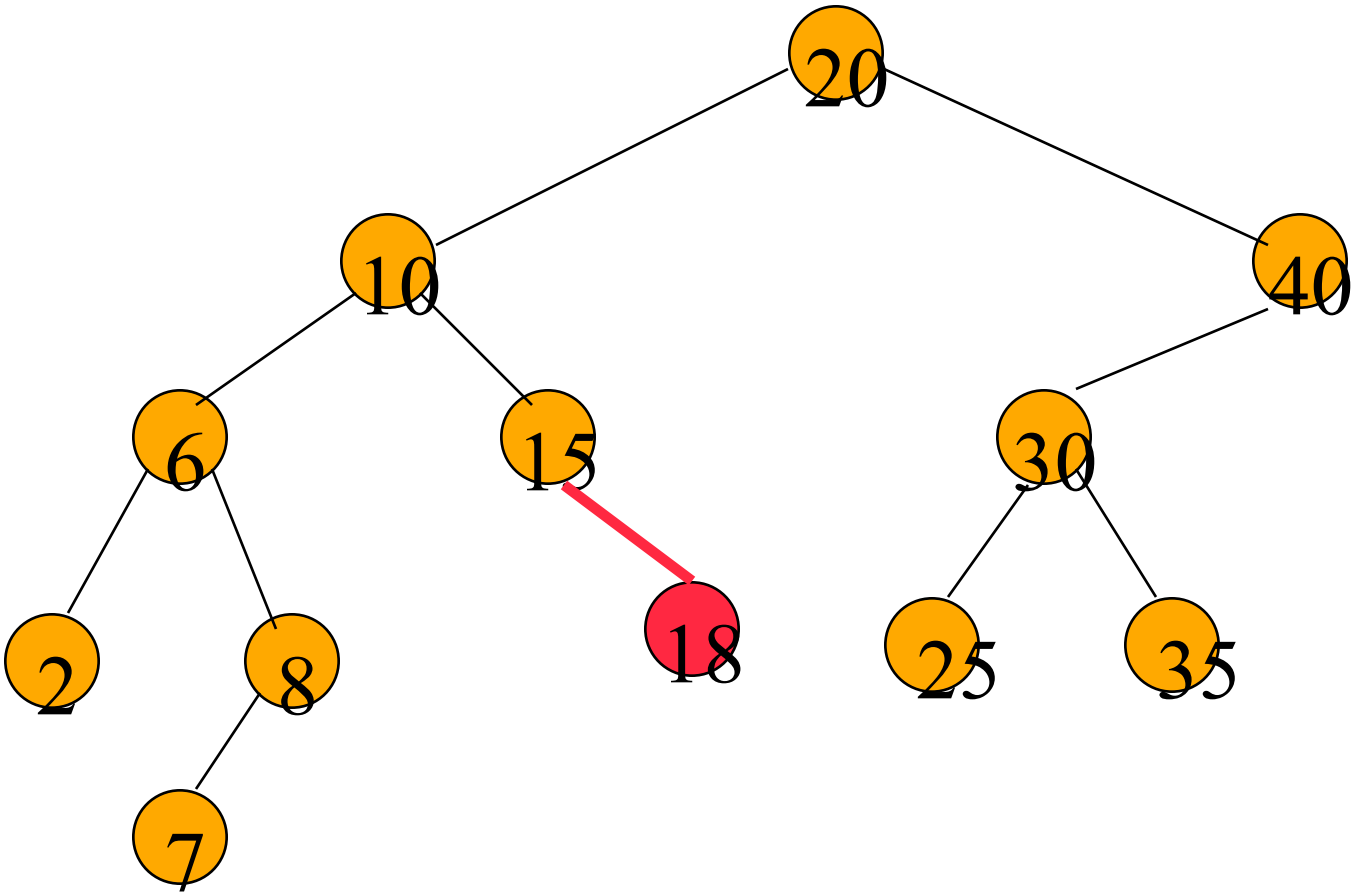
()


20
10
40
6
15
30
2
8
25
35

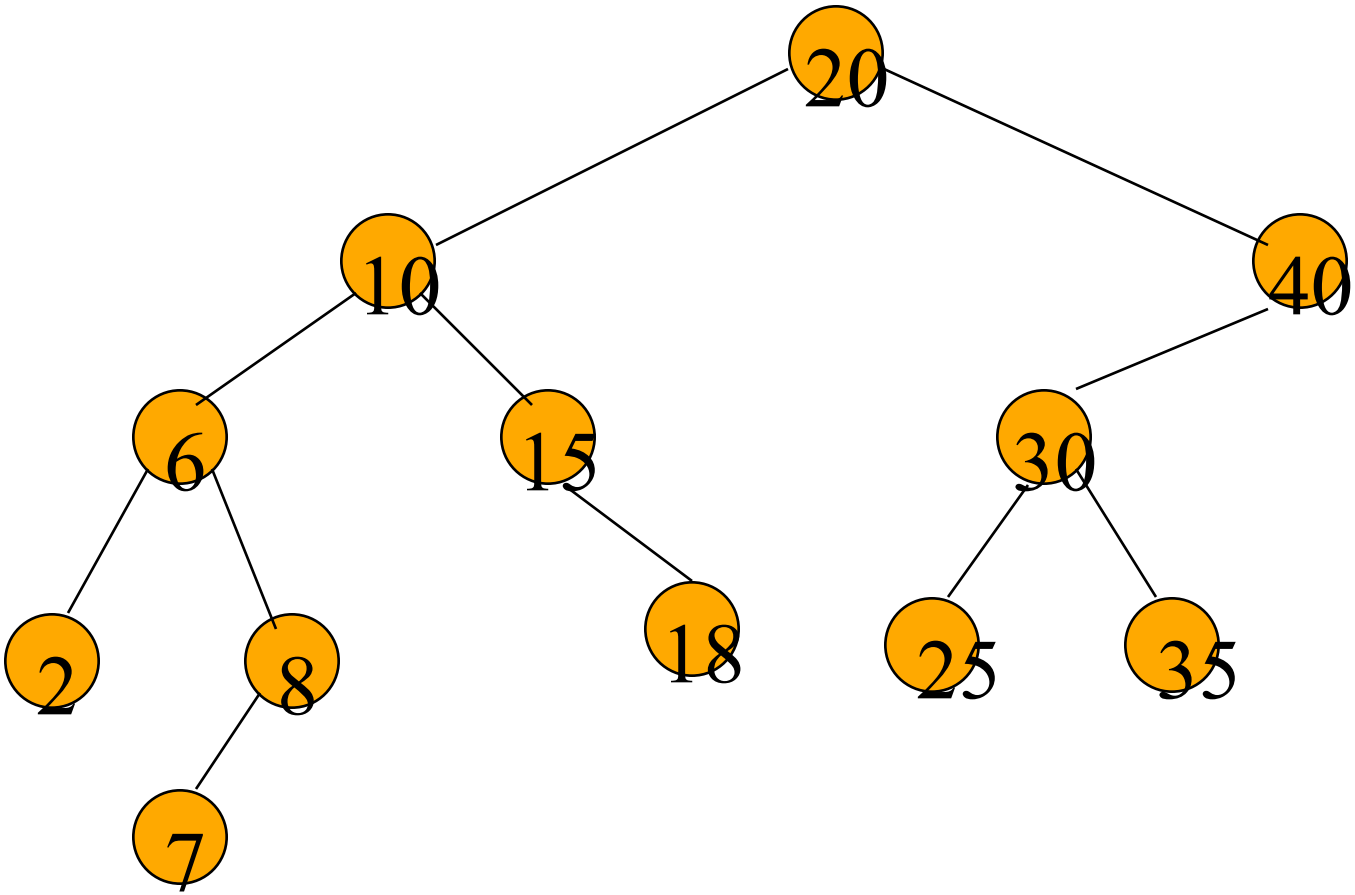
()



()


20
10
40
6
15
30
2
8
18
25
35
7

()


20
10
40
6
15
30
2
8
18
25
35
7

() ( ).

< , >

< , >:: (< , >&)

< < , >> * = , * =0

( )

=

(. < → .) = →

(. > → .) = →

// ,

→ . = .

= < < , >>(,0,0)

( ) //

(. < → .) → =

→ =

= ( )

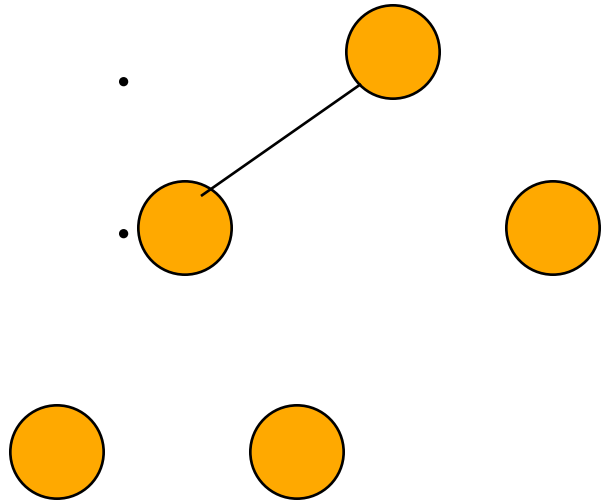
()

:

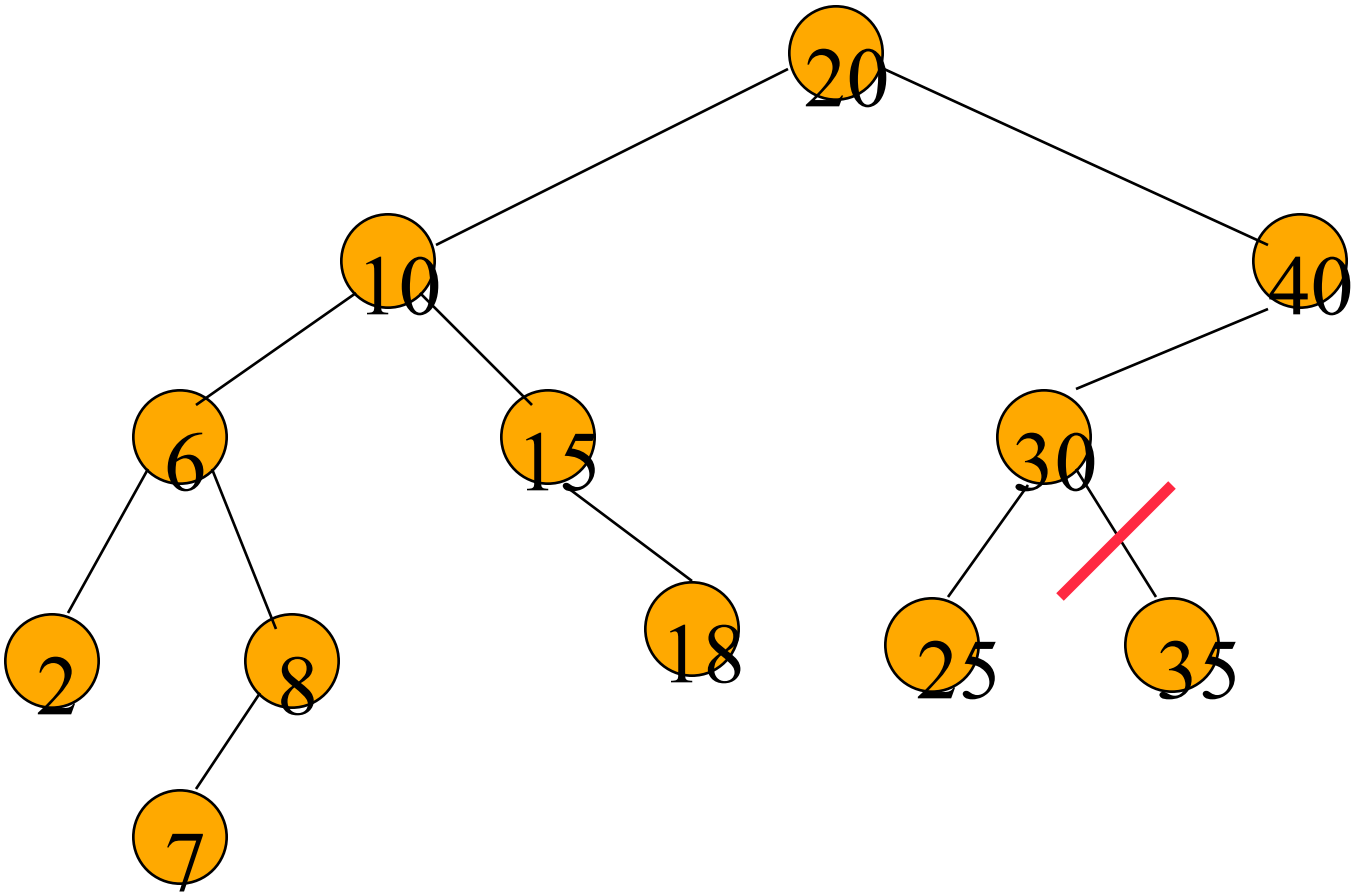
.

1

2

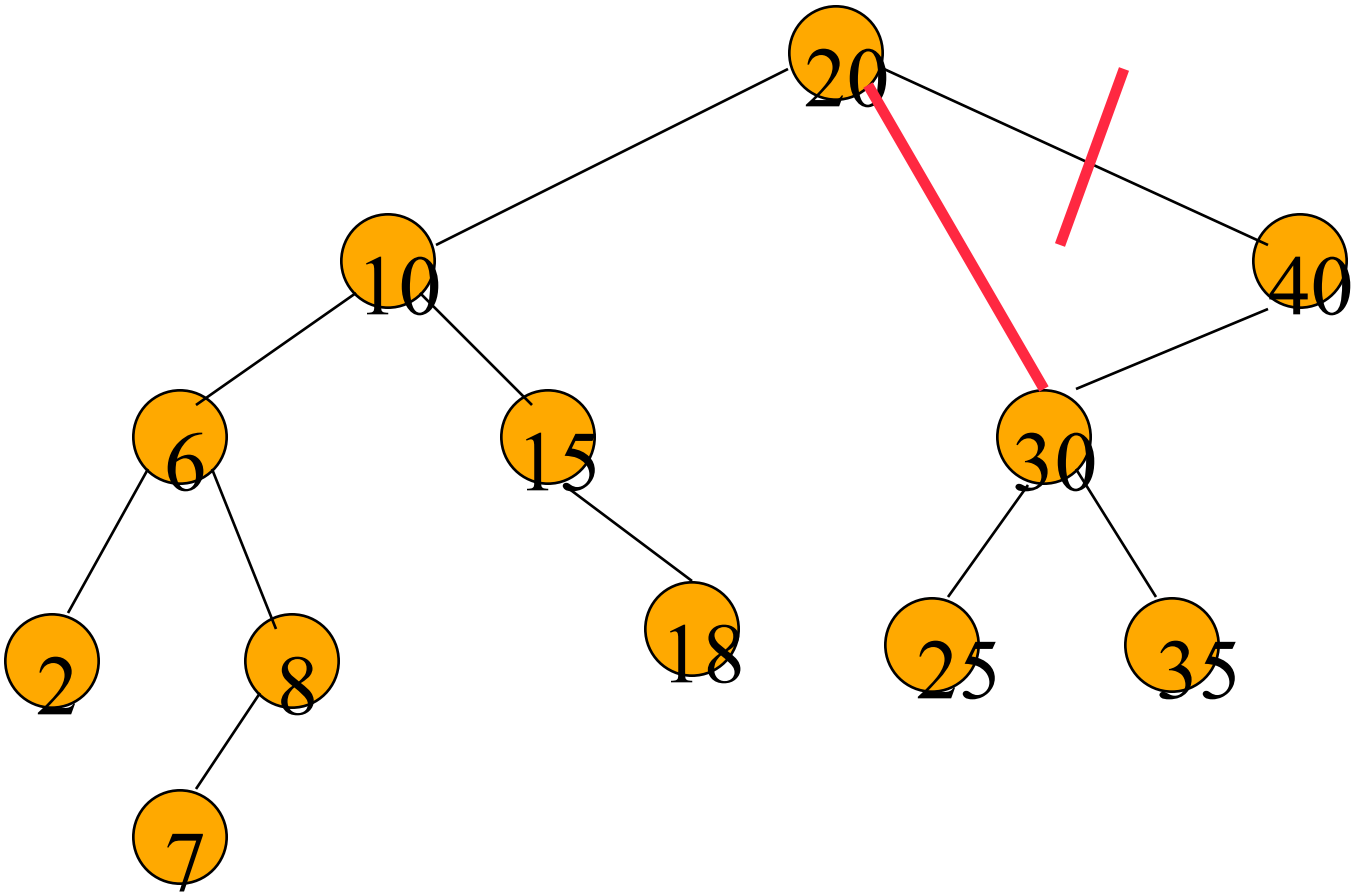


(.)



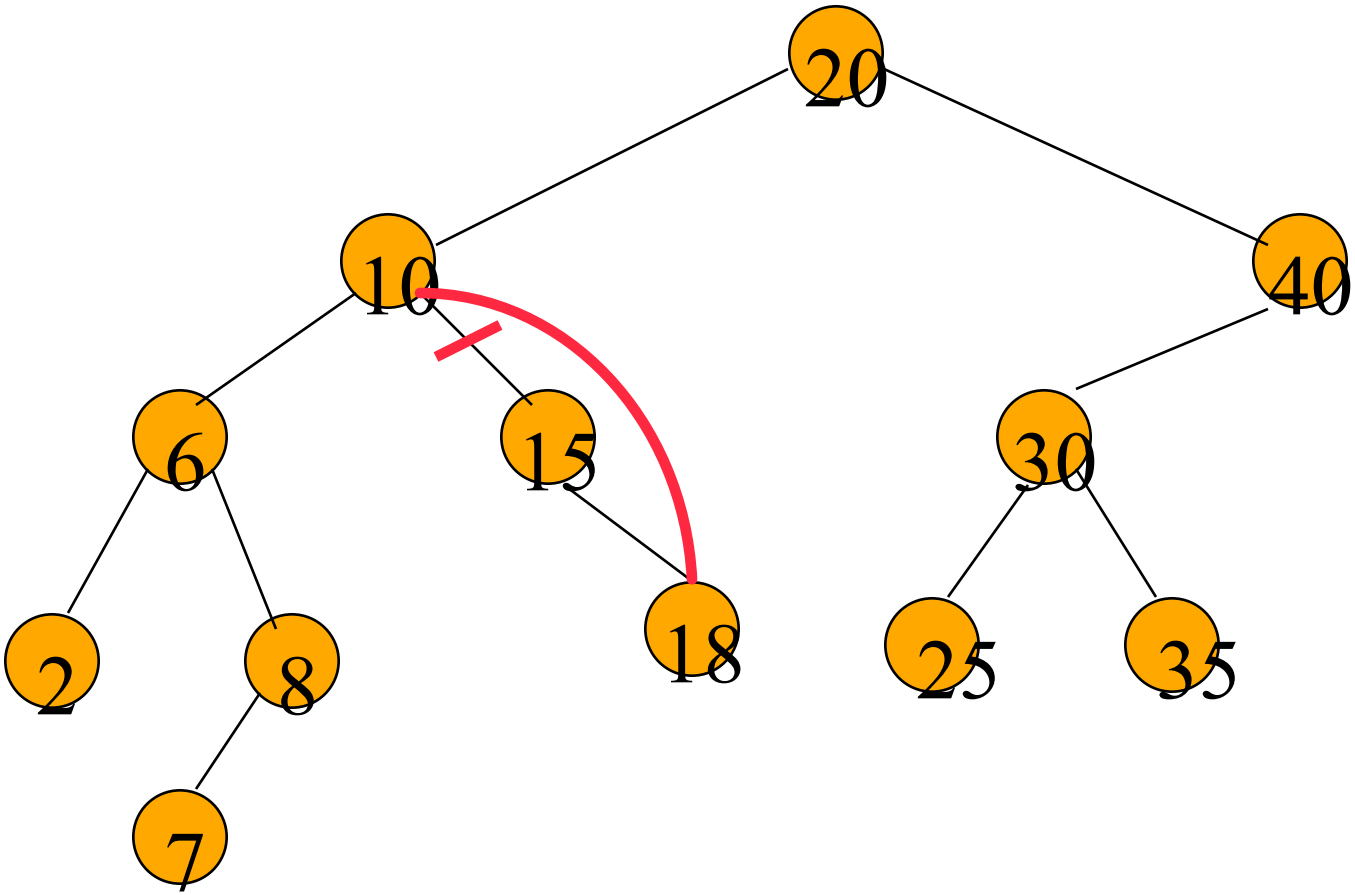
. = 35

1



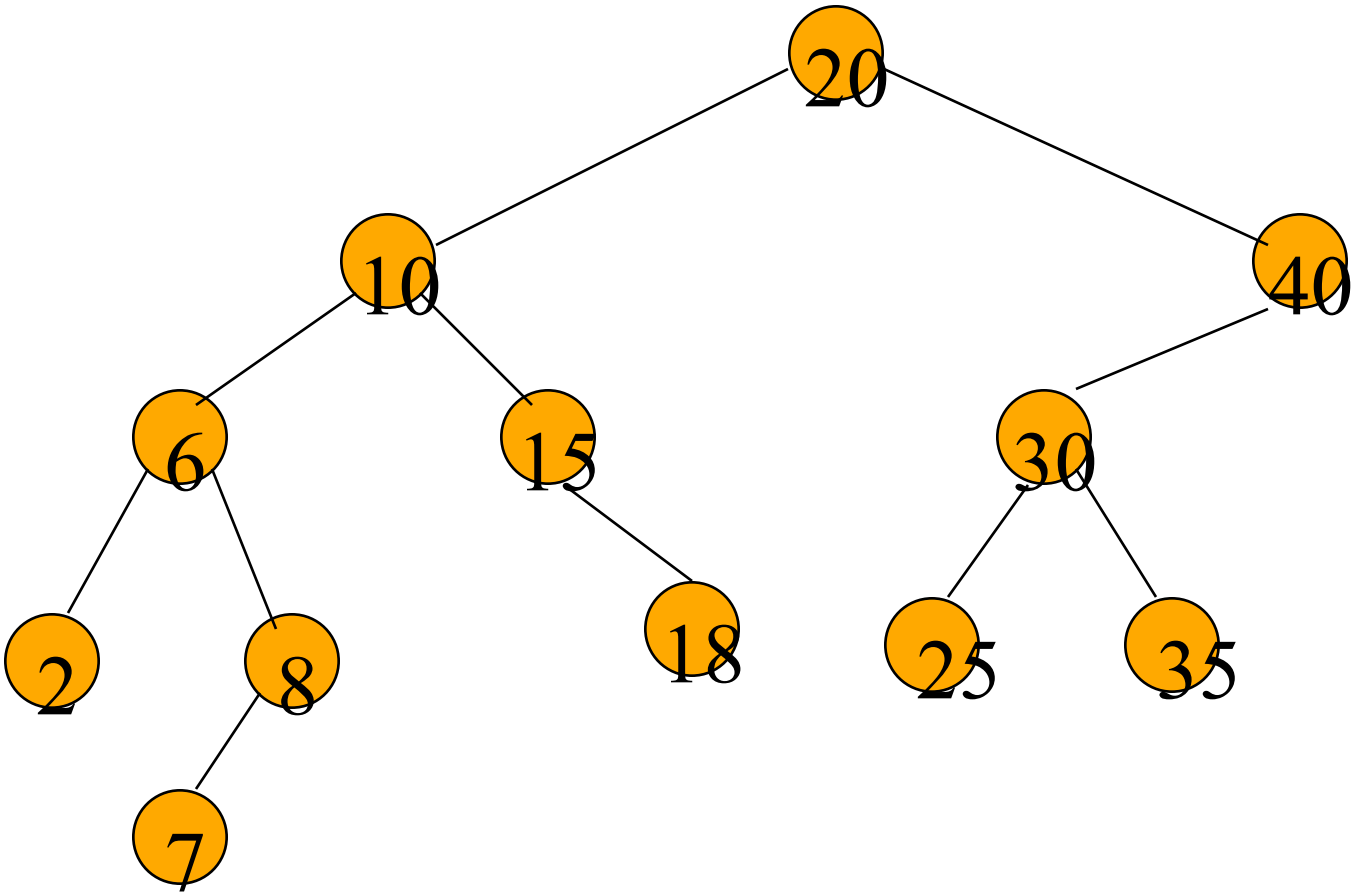
1 . = 40

1 (.)



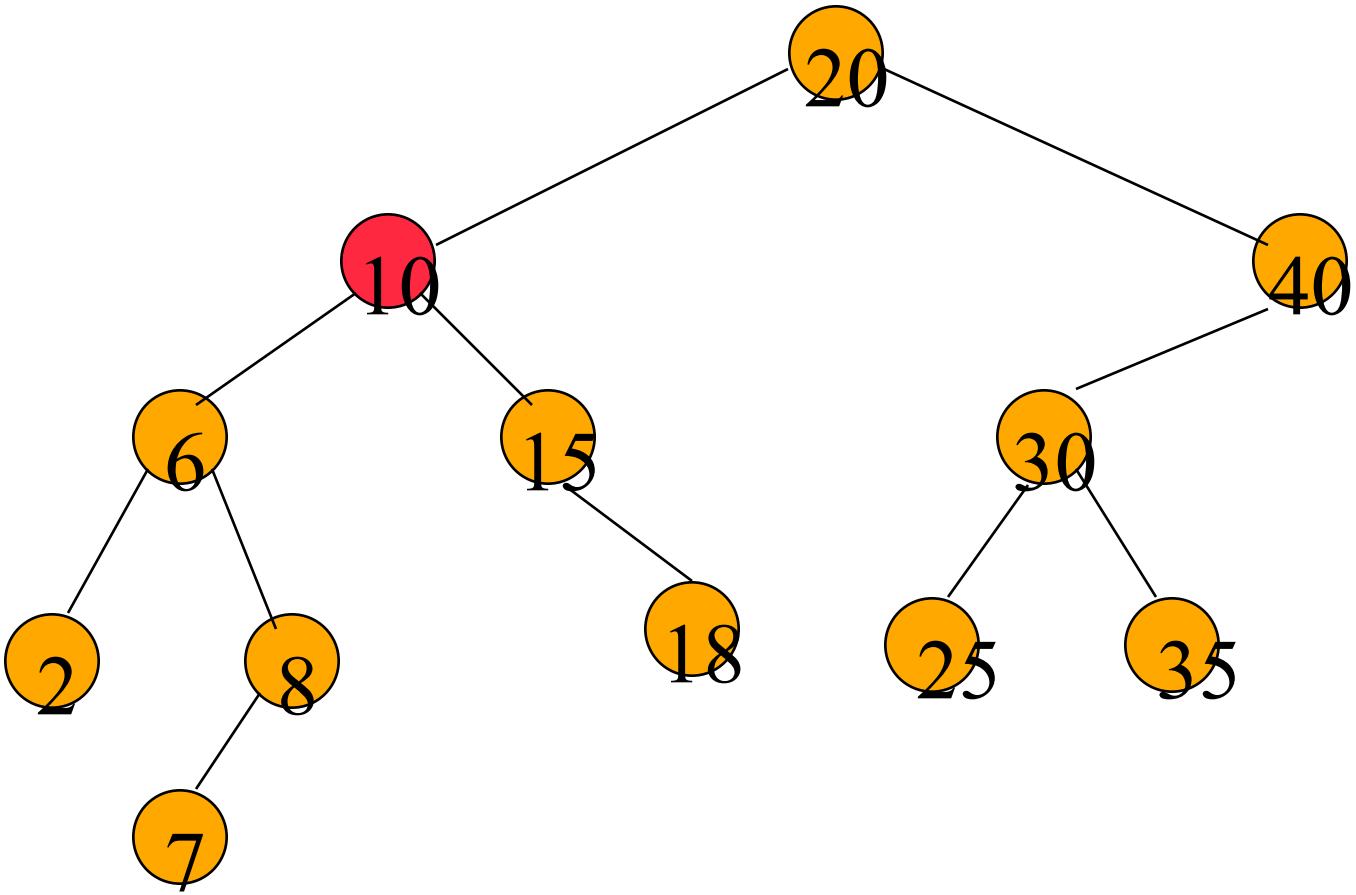
1 . = 15

2



2 . = 10

2



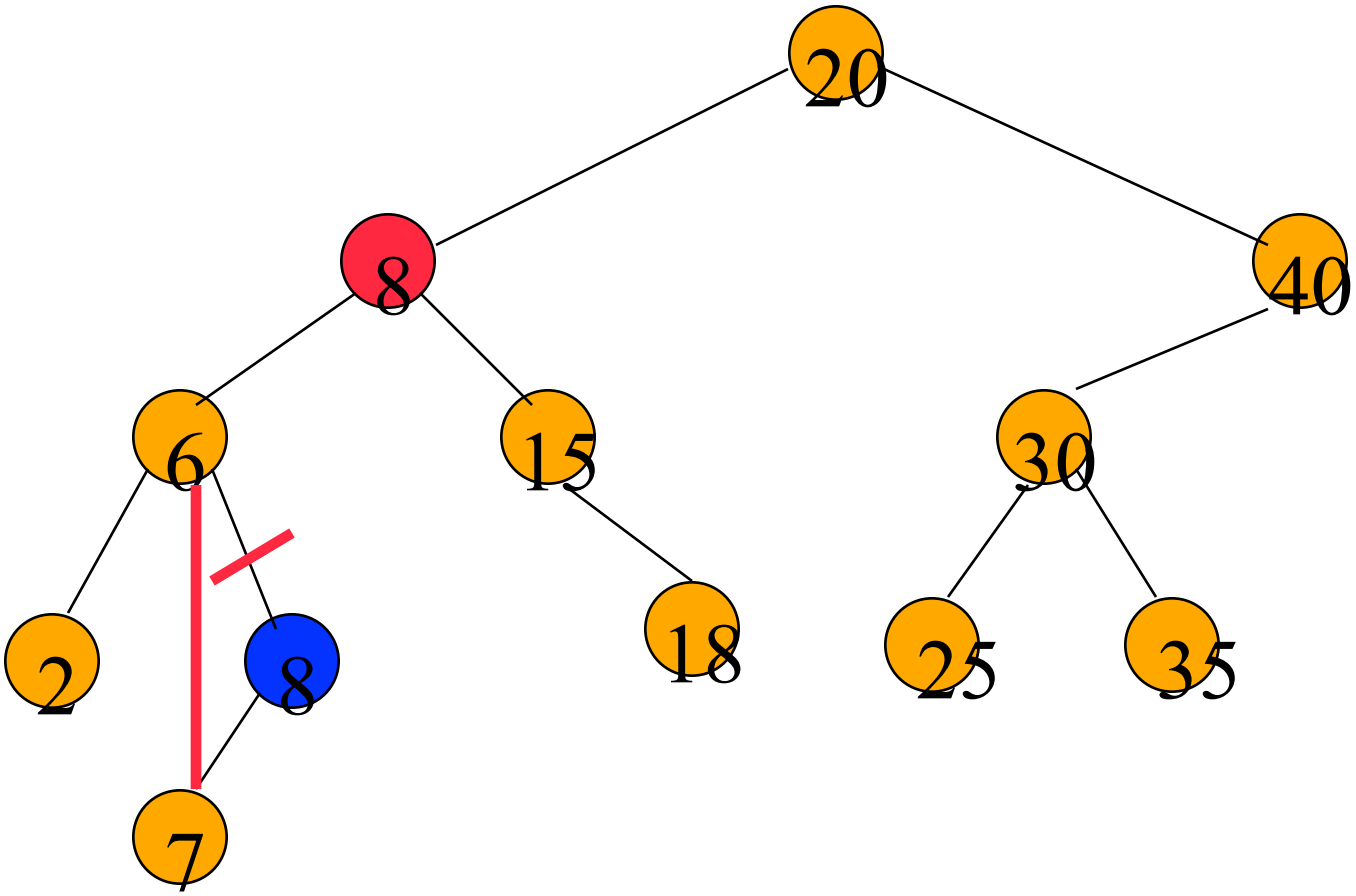
(

).

2

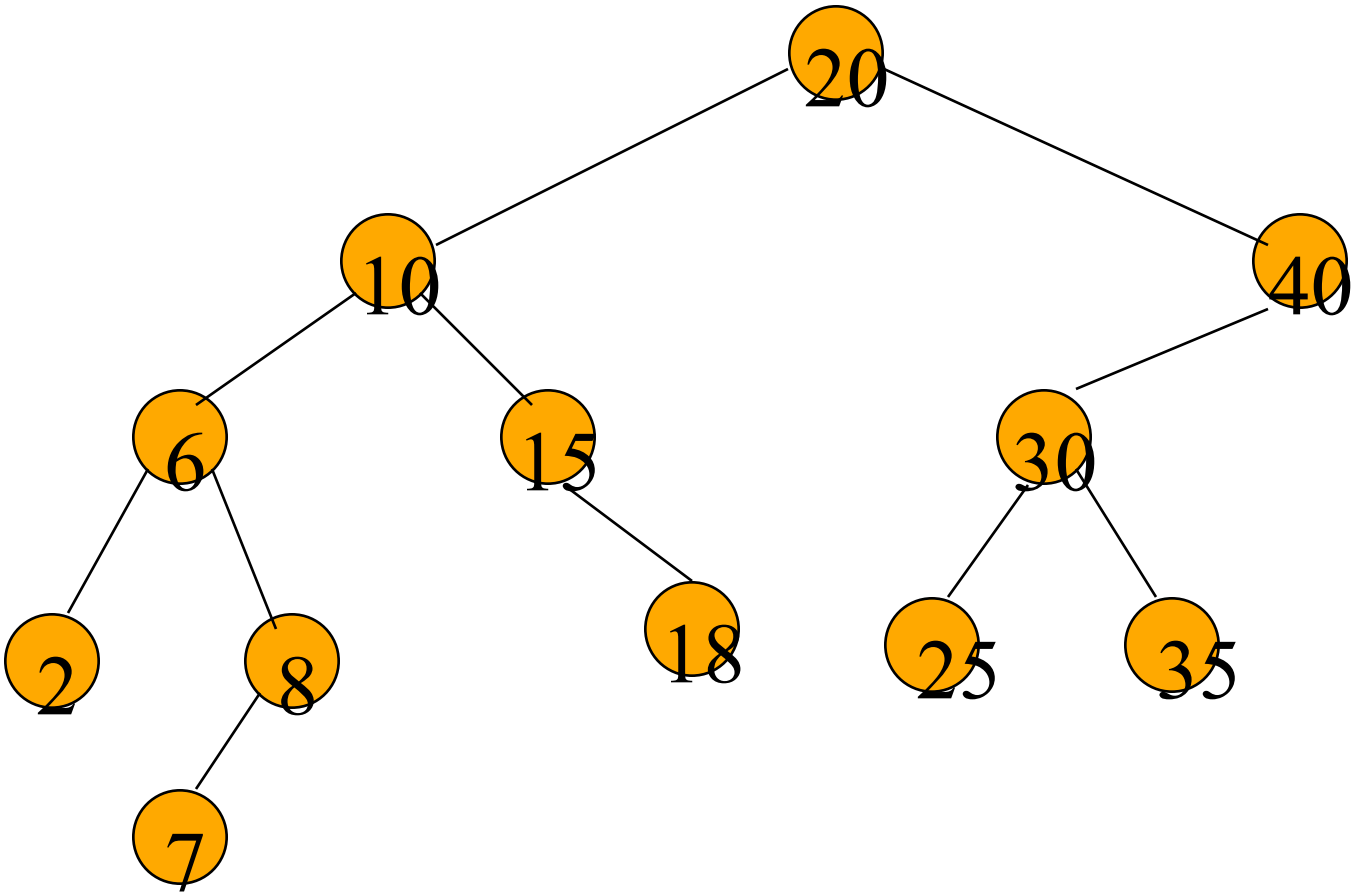
2

2

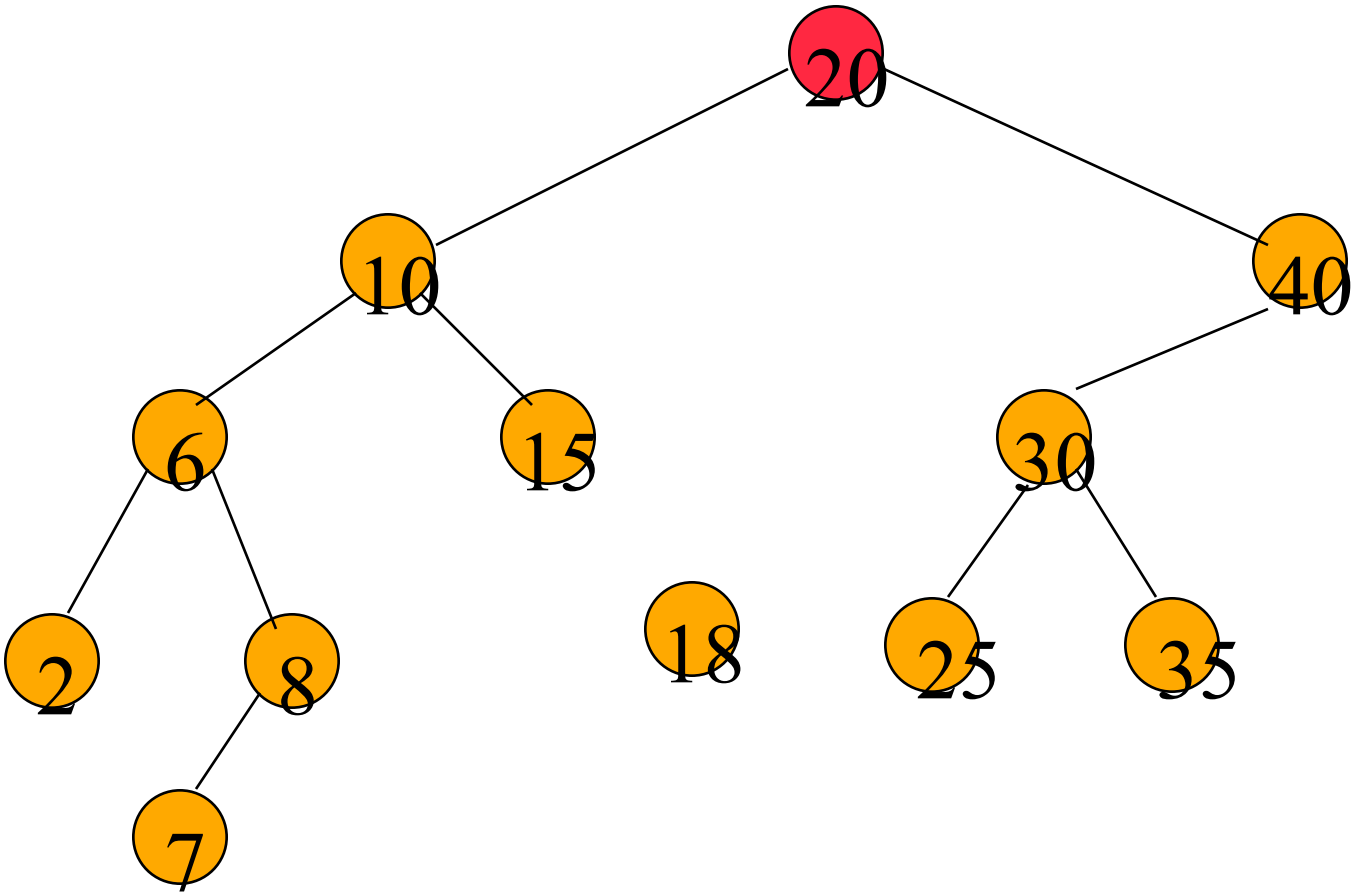


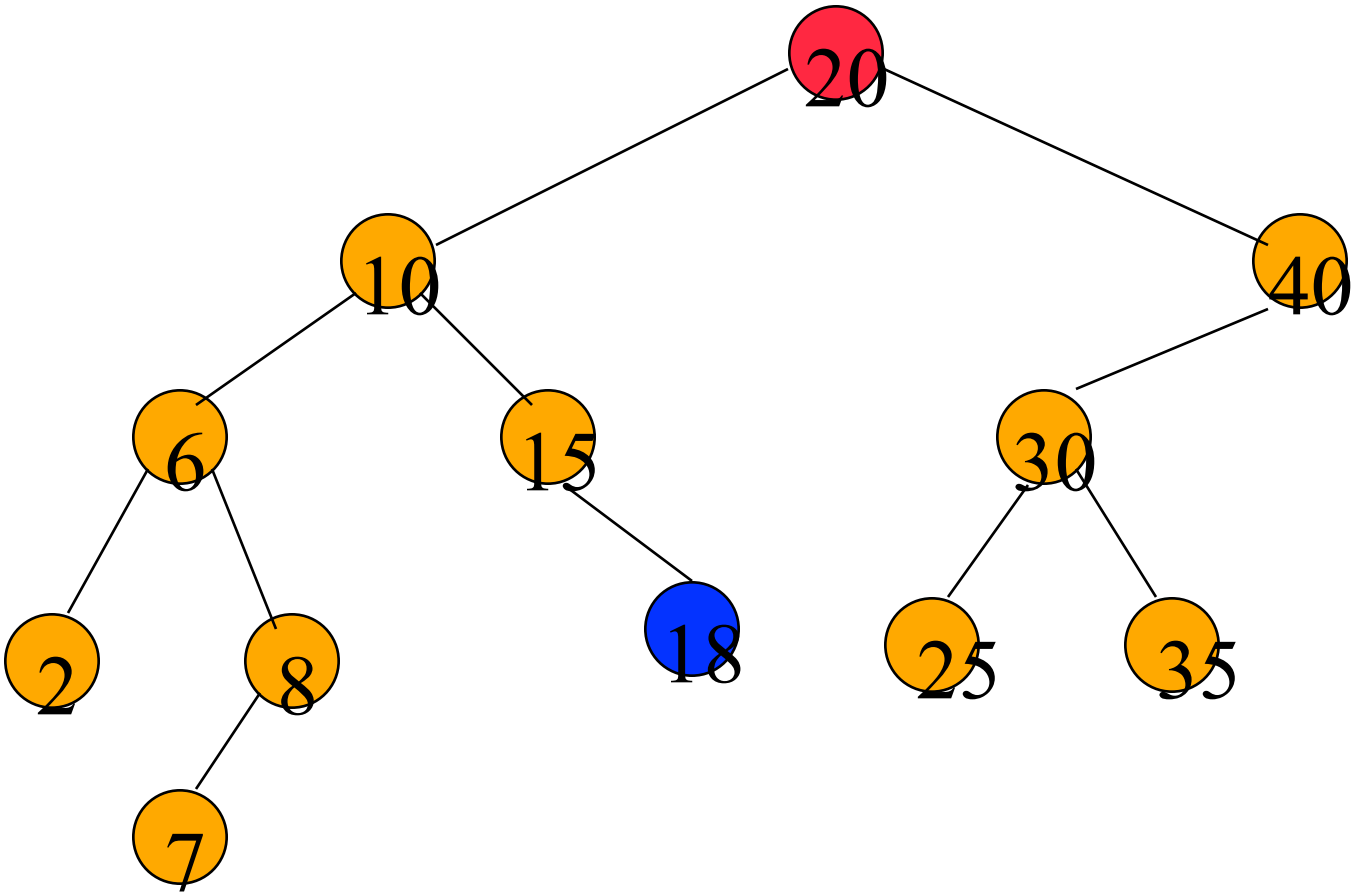
1 .

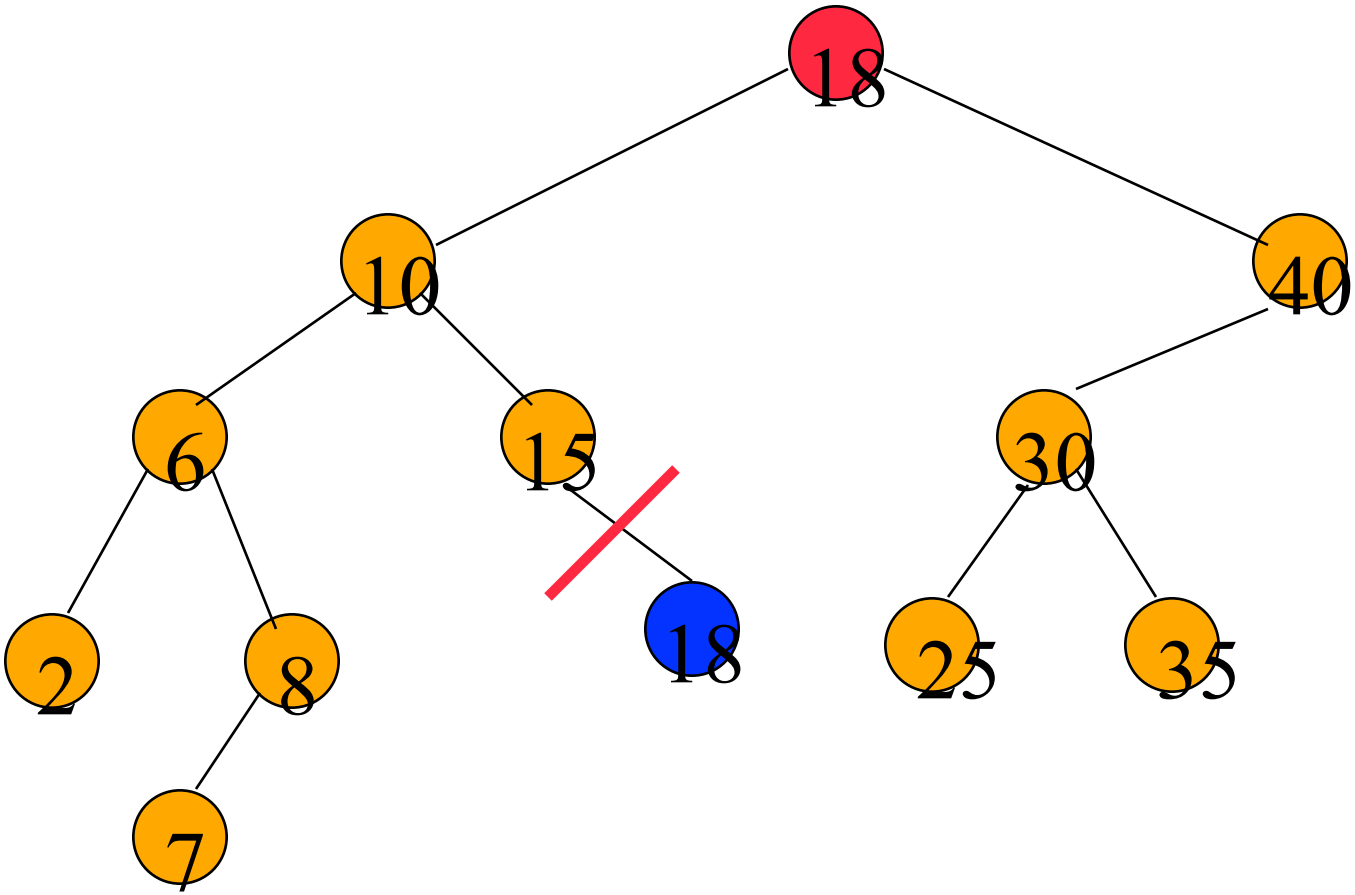
2



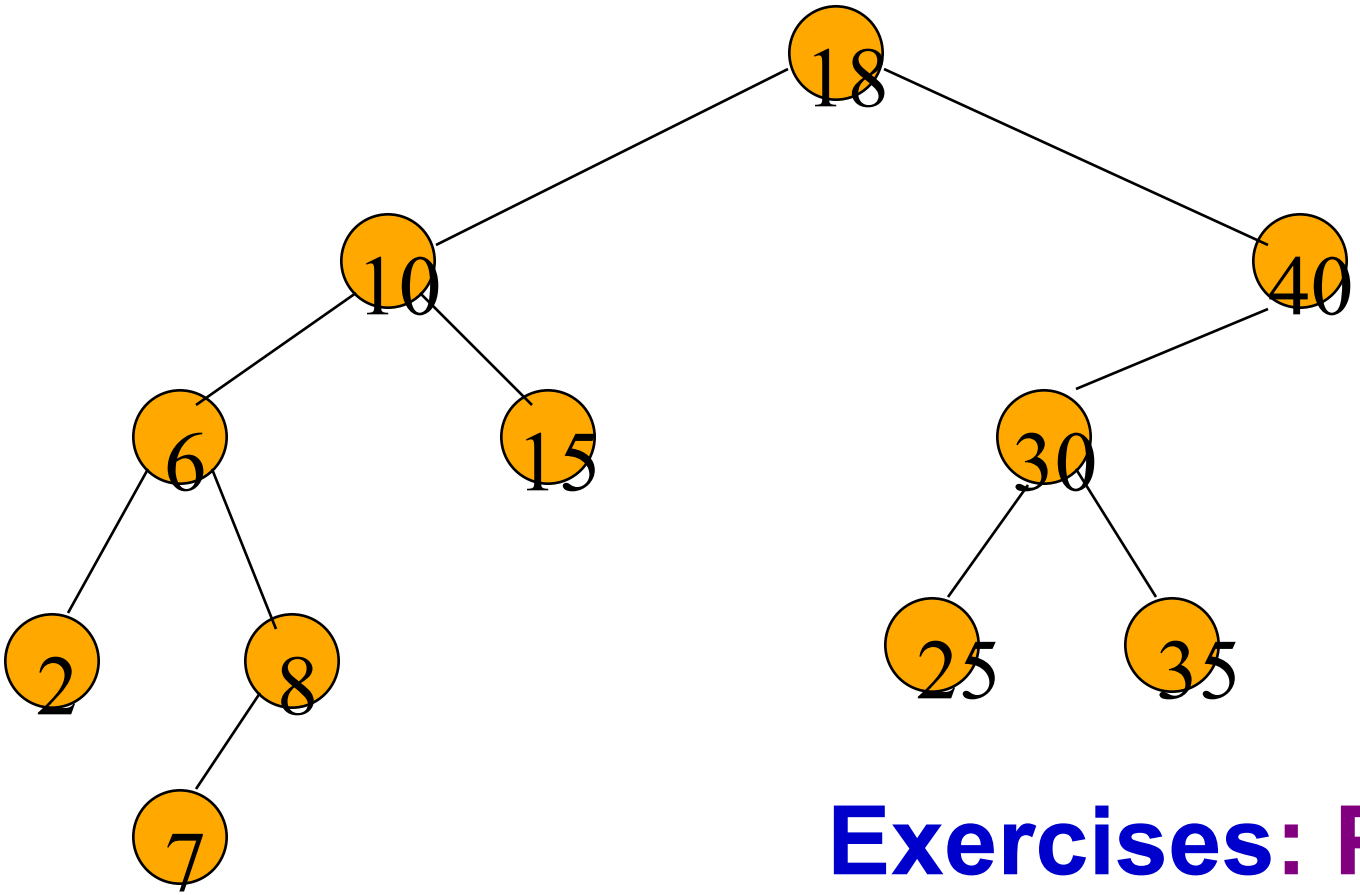
2 . = 20


20
10
40
6
15
30
2
8
18
25
35
7


20
10
40
6
15
30
2
8
18
25
35
7

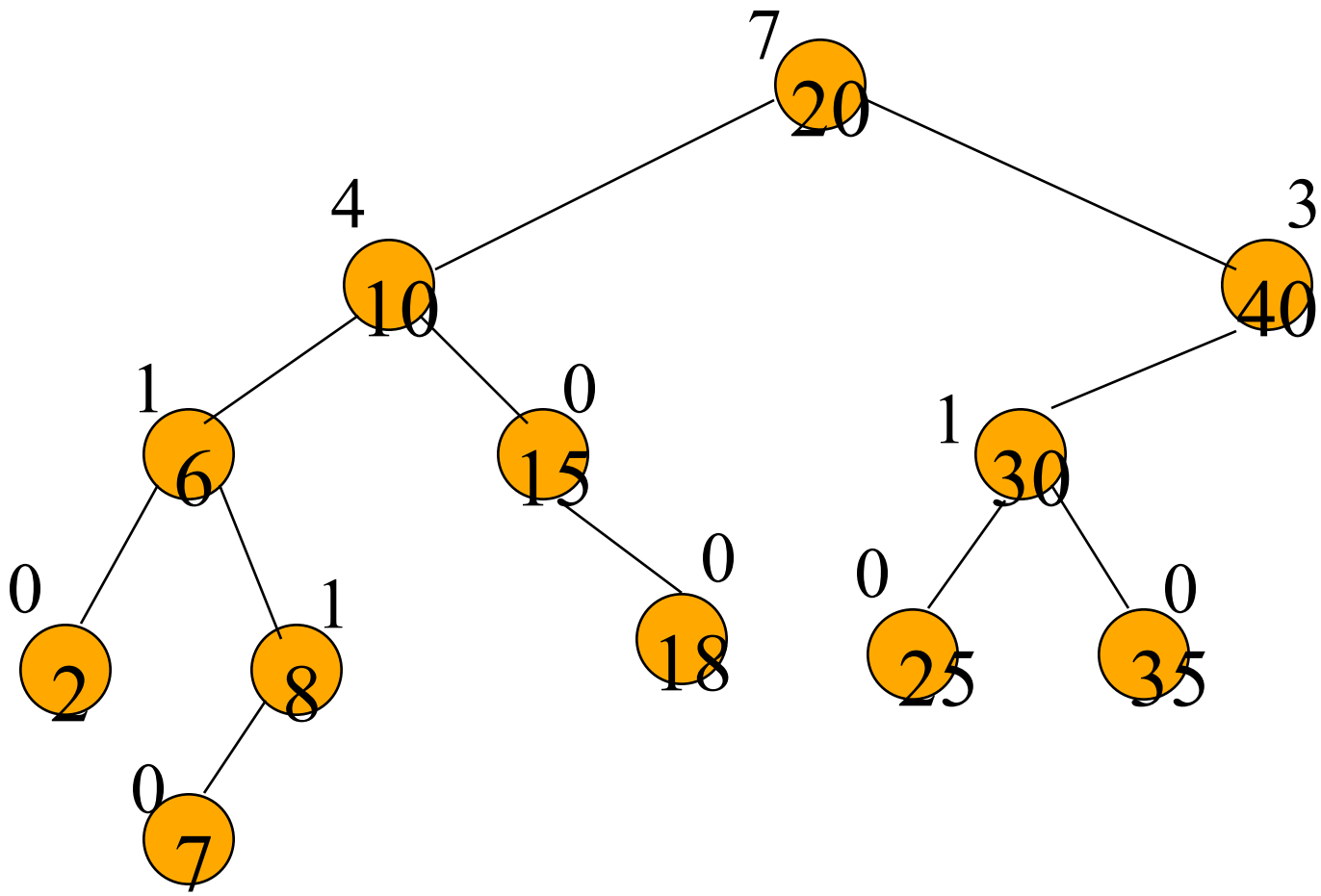

18
10
40
6
15
30
2
8
18
25
35
7

2



Exercises: P296-1,2

( ).


7
20
4
10
3
40
1
6
0
15
1
30
0
2
1
8
0
18
0
25
0
35
0
7

(=).

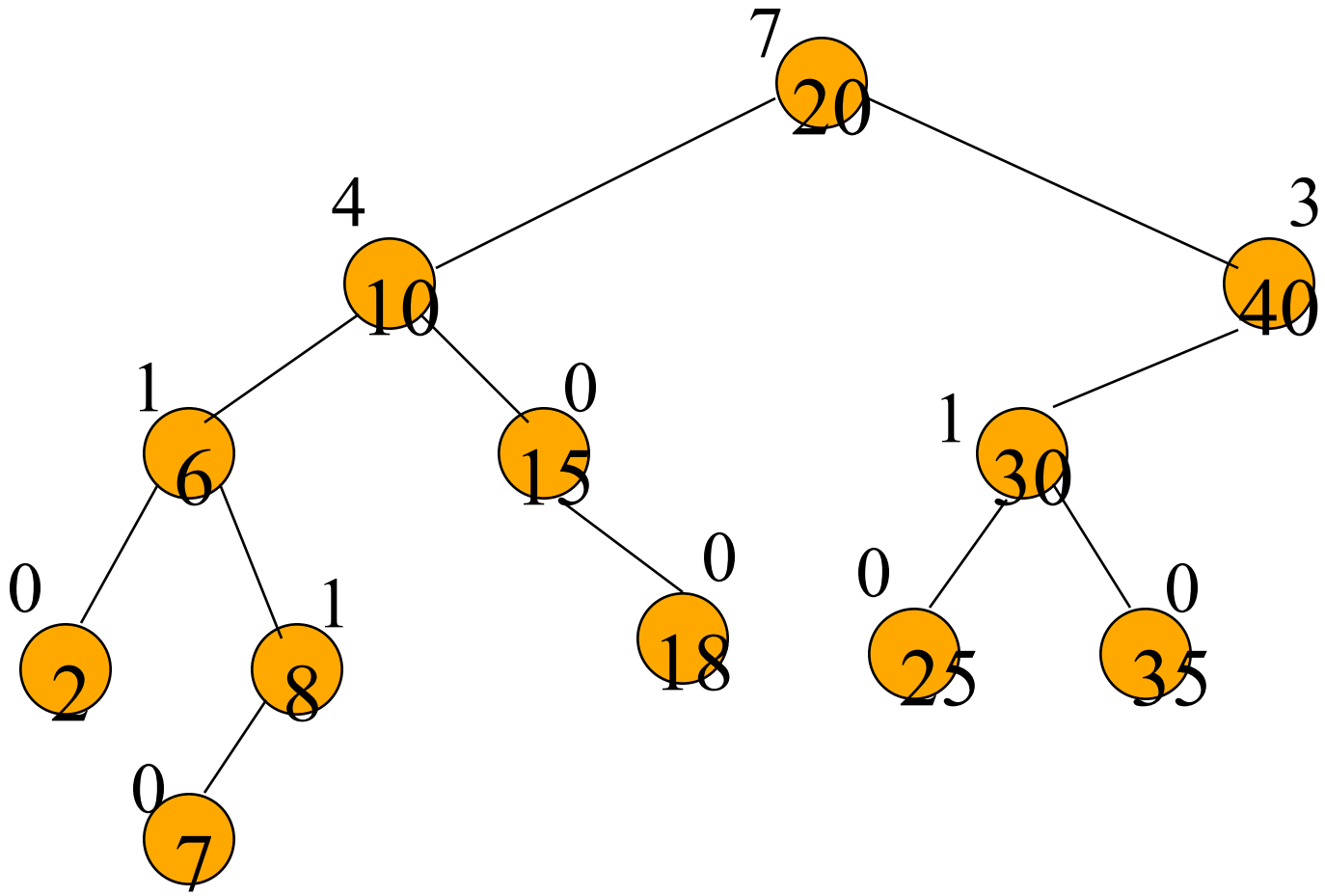
2,6,7,8,10,15,18,20,25,30,35,40

(2) = 0

(15) = 5

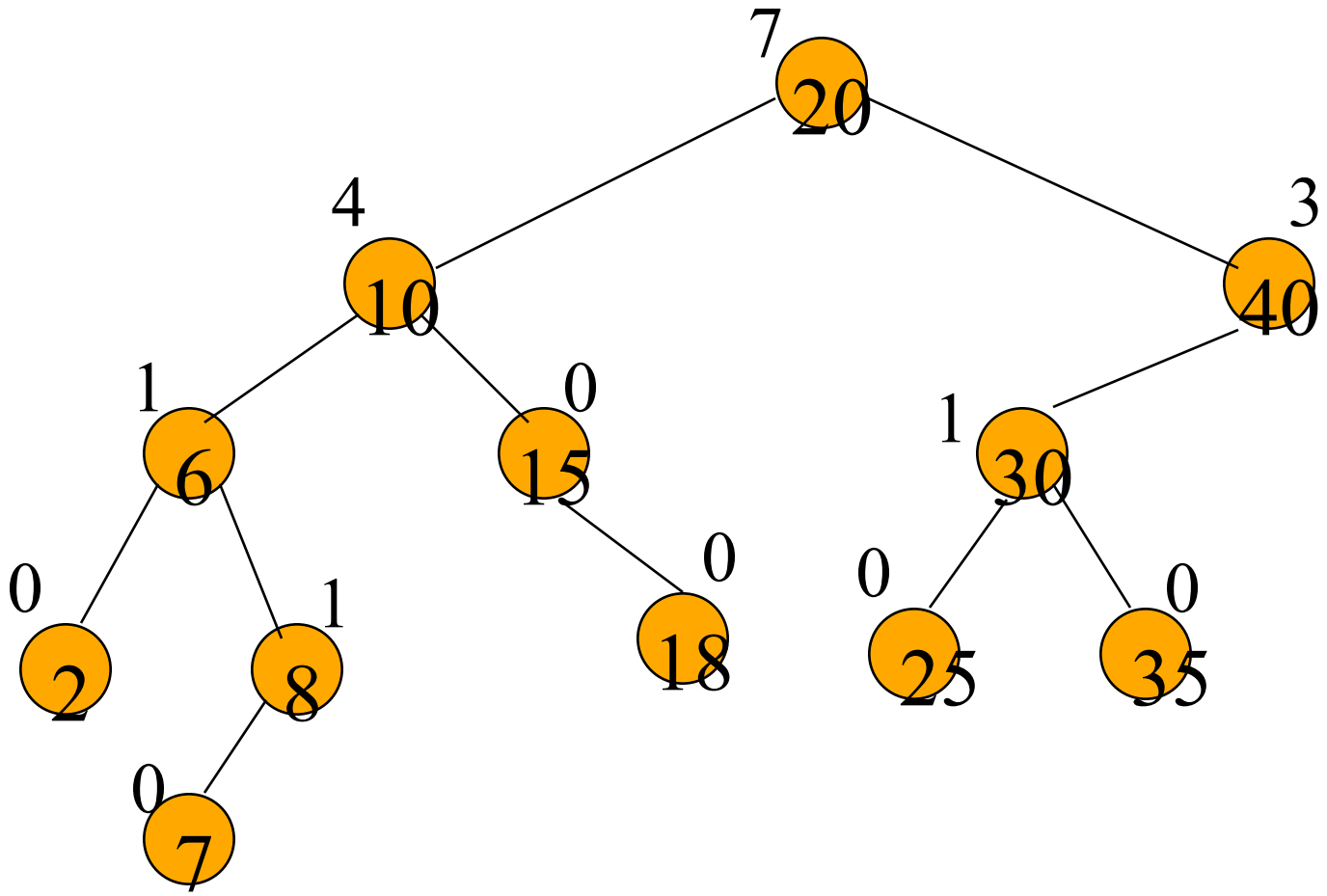
(20) = 7

( ) = ( )



= 2,6,7,8,10,15,18,20,25,30,35,40

( ) ( )



= 2,6,7,8,10,15,18,20,25,30,35,40

( ) ( )

=

.

<

.

>

.

(- . -1)

( )

- ( ) .

( ),

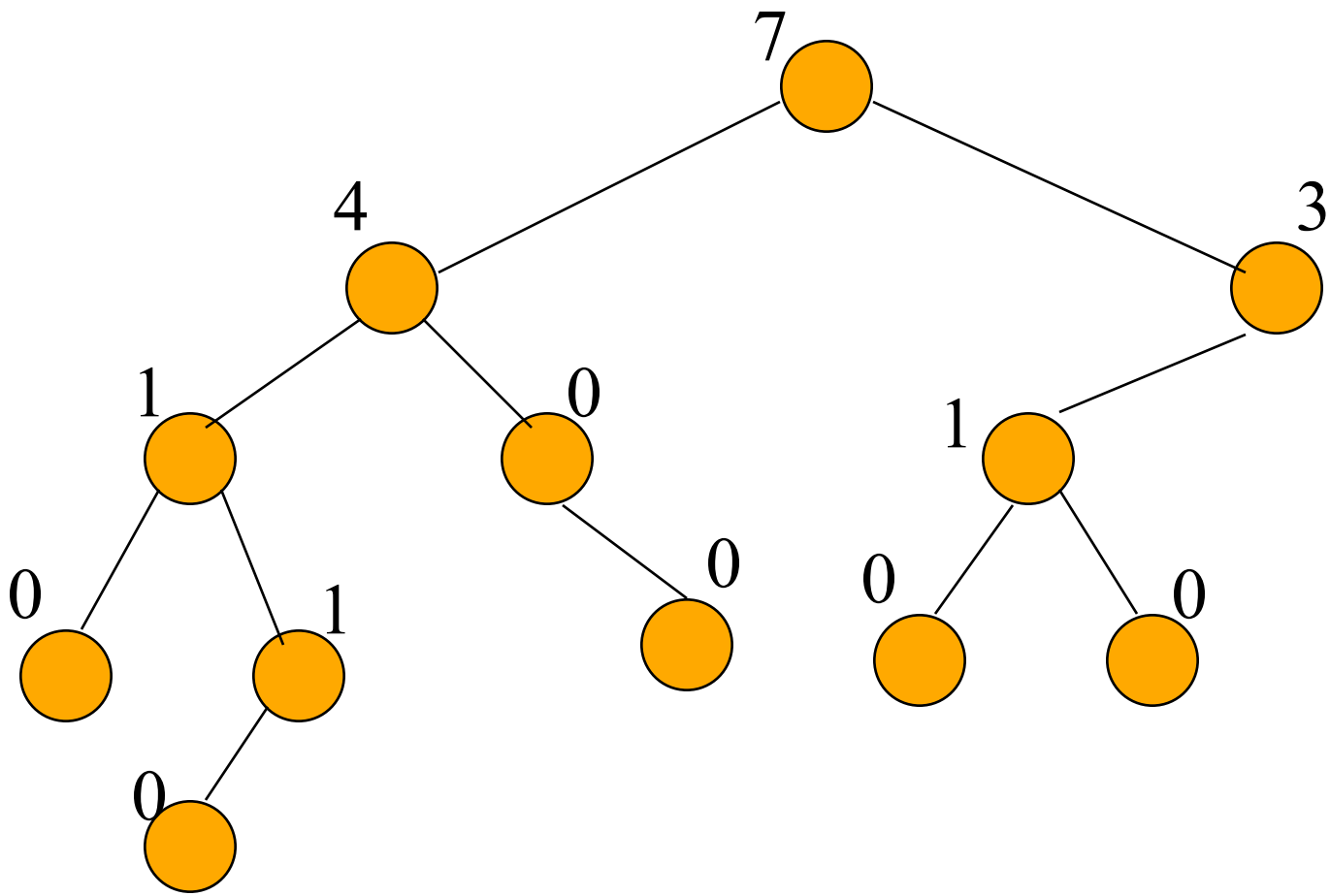
(,), ( )

(( )) (

,

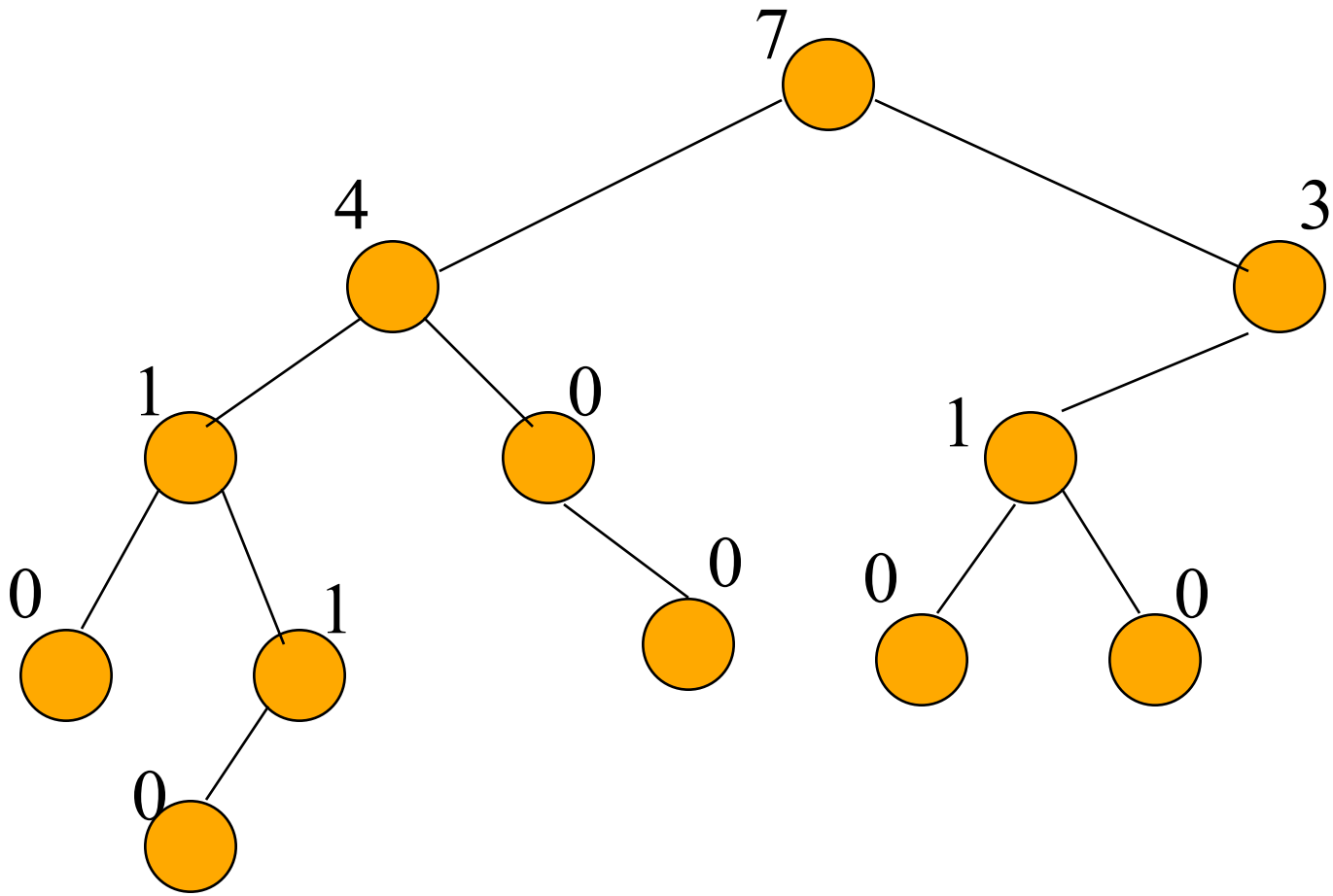
).

.


7
4
3
1
0
1
0
0
0
1
0
0

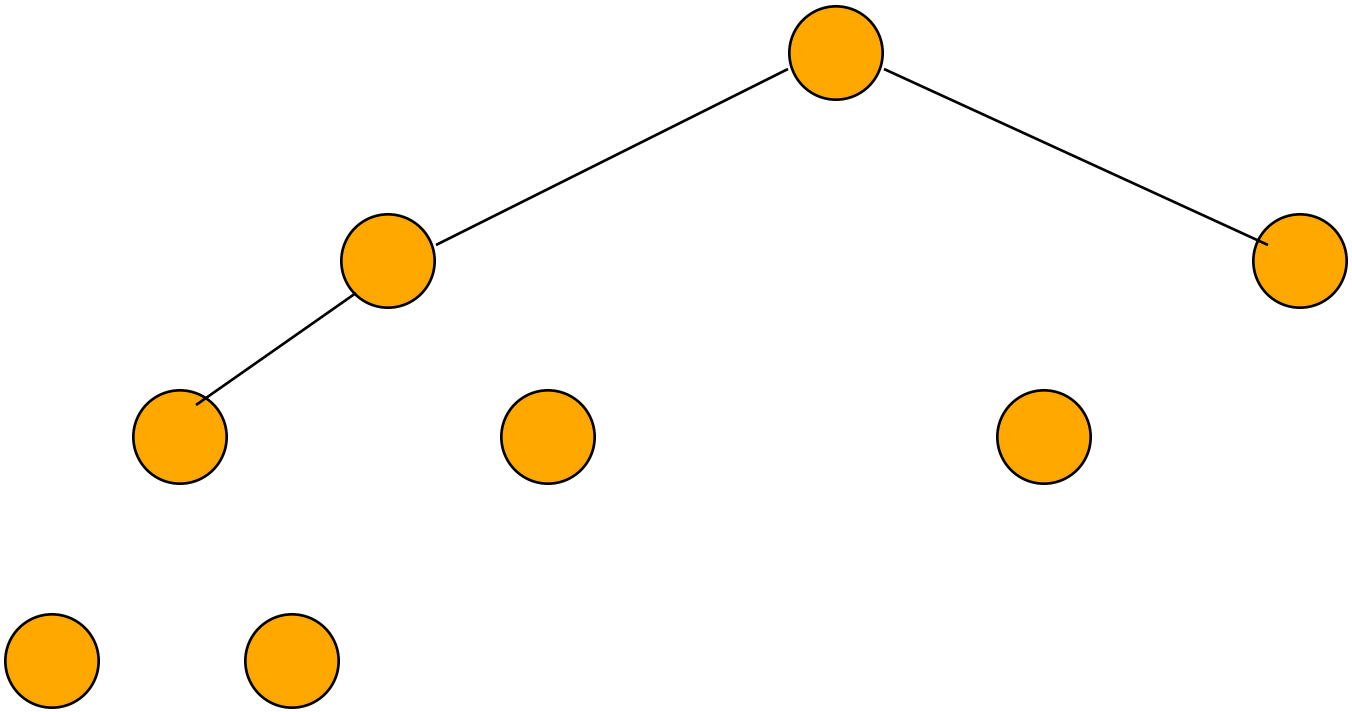
= , , , , , , , , , , ,

(5,)

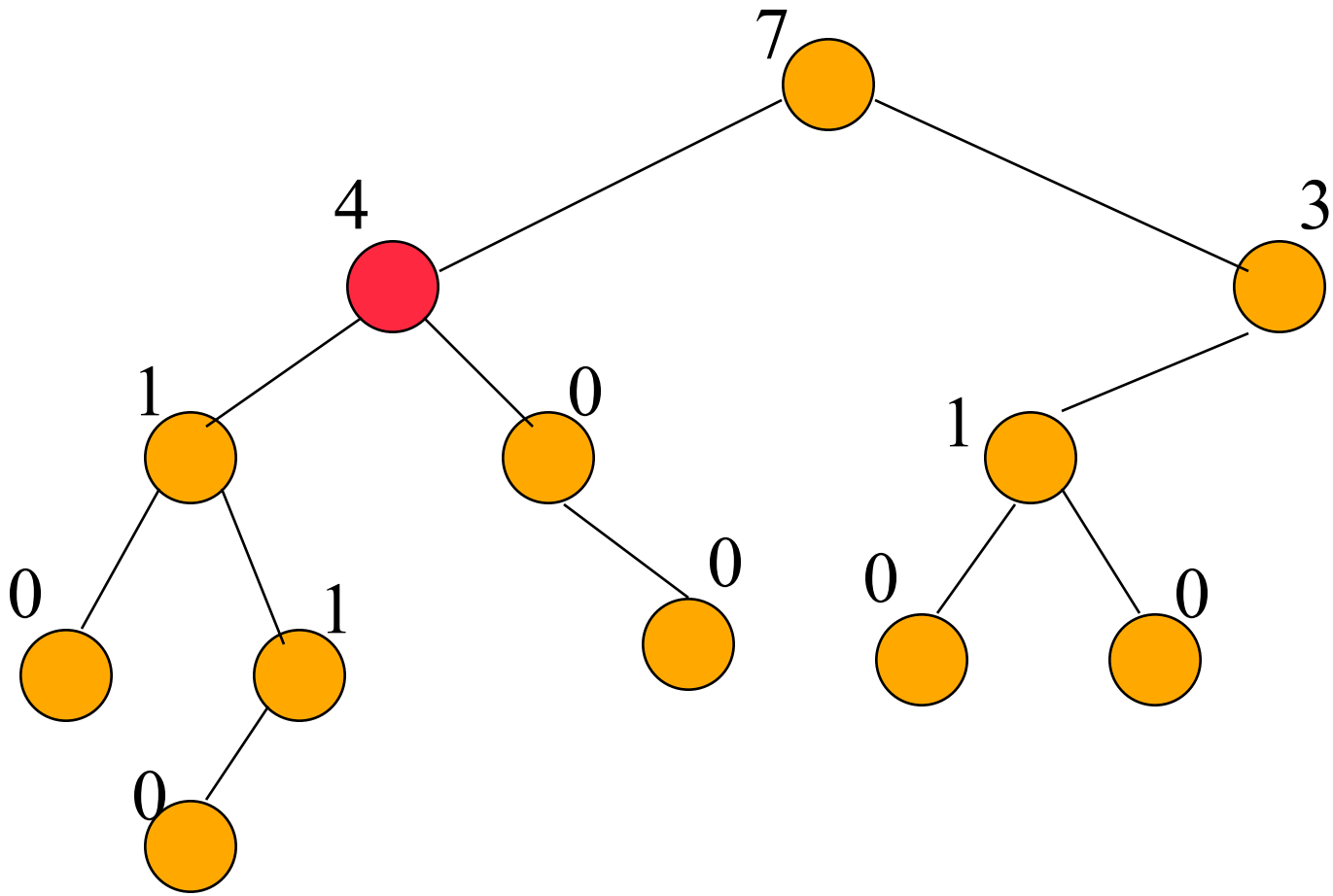


= , , , , , , , , , , , ,

(5,)



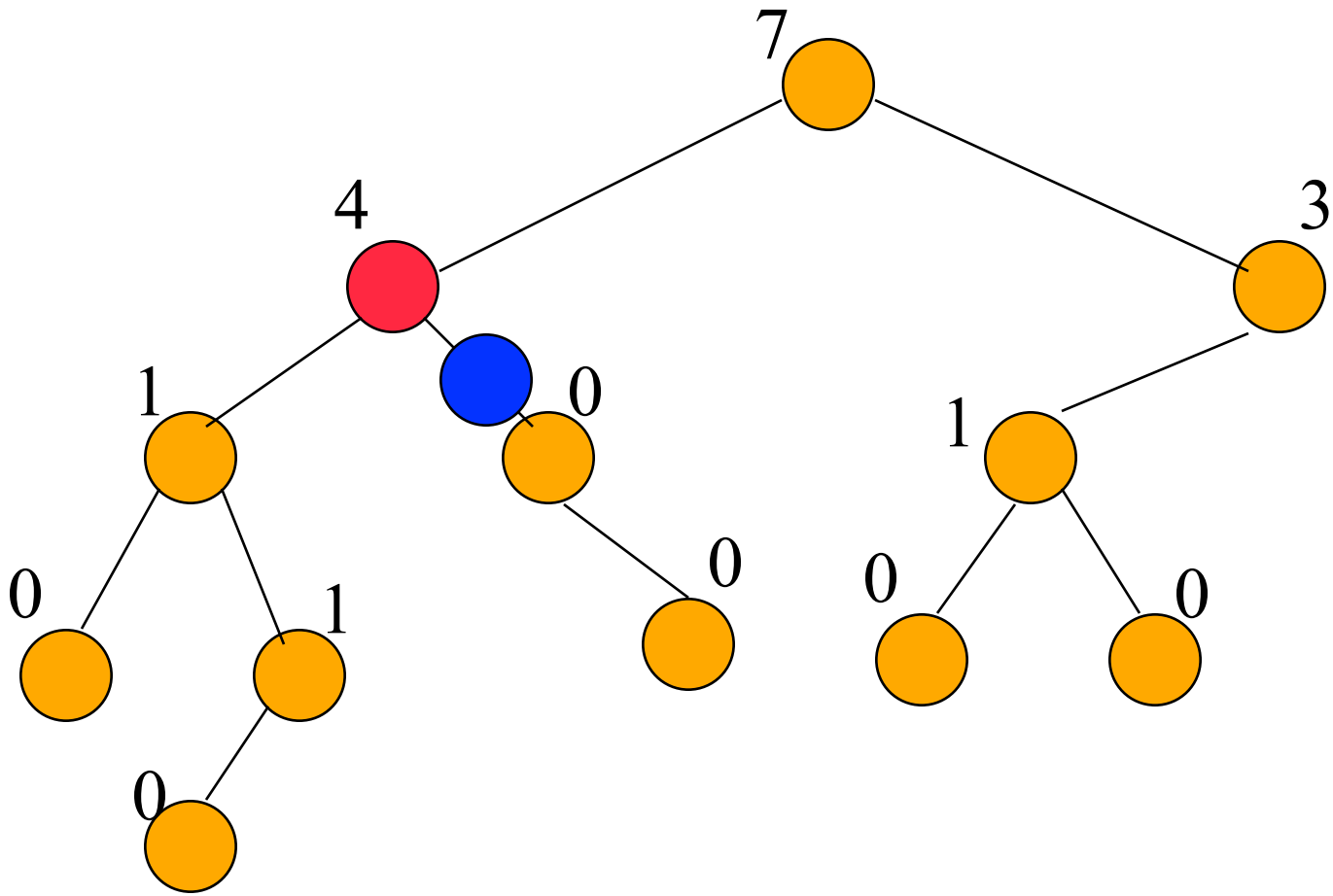
(5,)



= , , , , , , , , , , ,

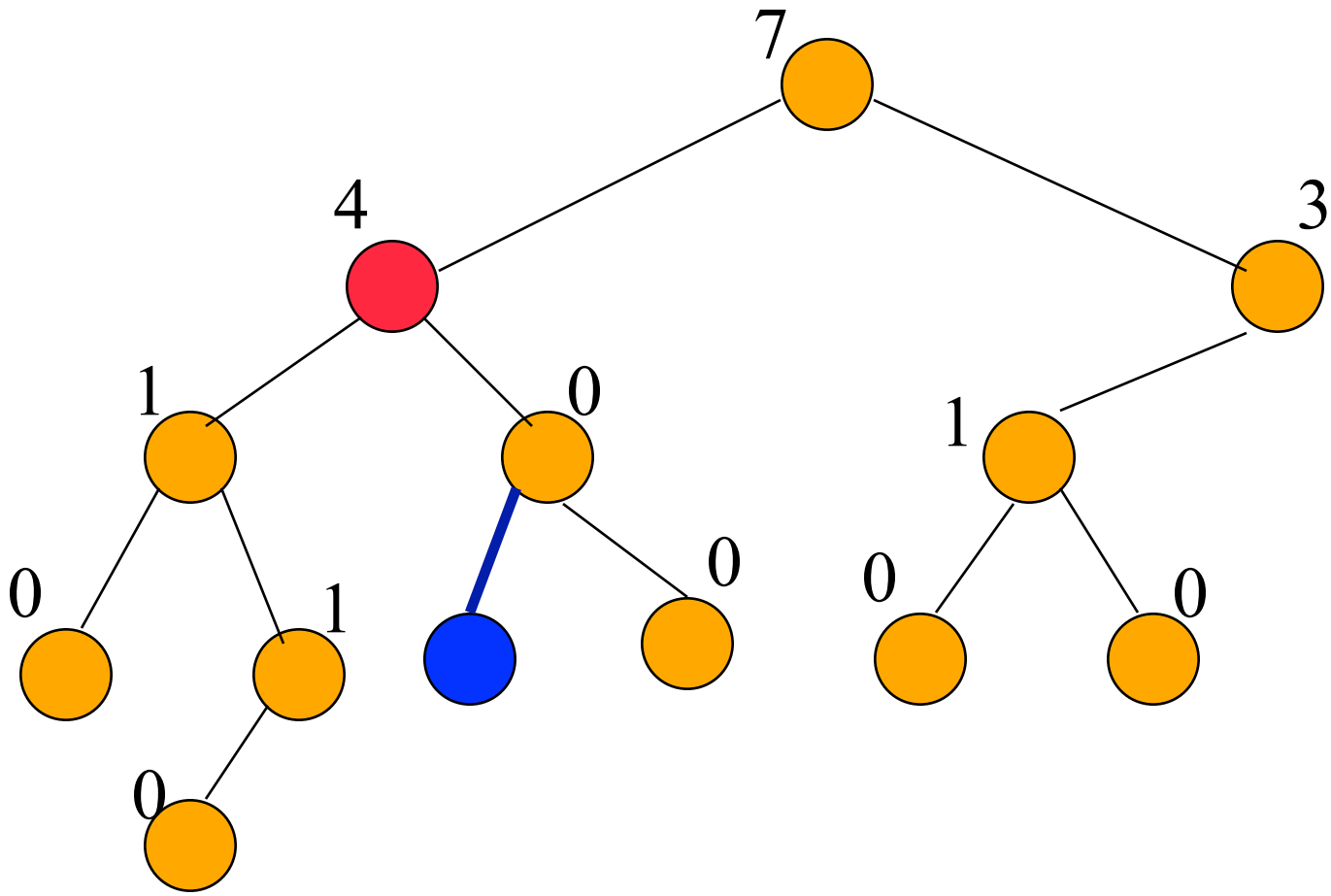
4 ( )

(5,)



;

(5,)



(5,)

.

.

( ).

:

- :: ( );
- :: (,
  );

:

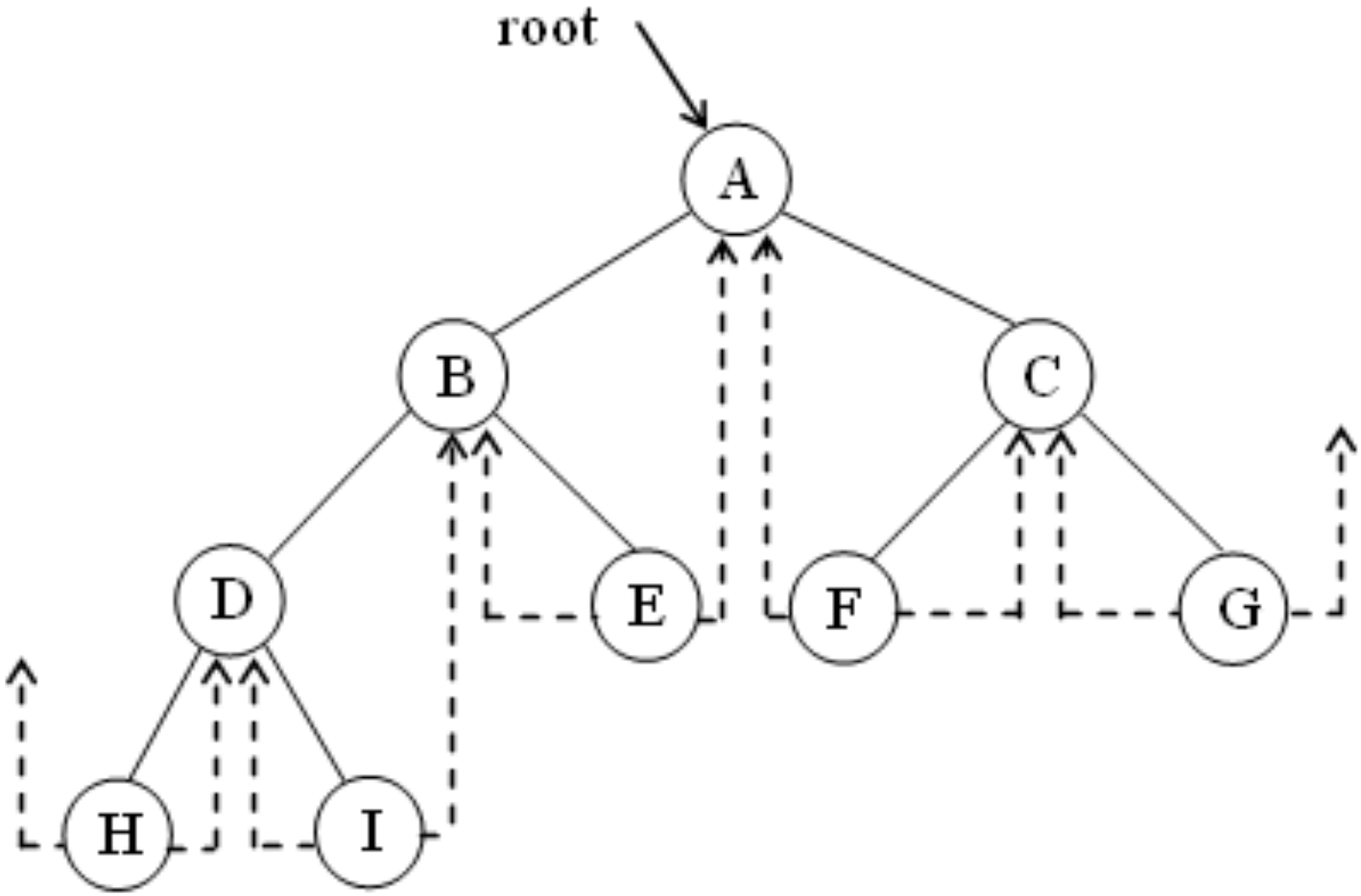
2

:

(1) 0

.

0


root
A
B
C
D
E
F
G
H
I

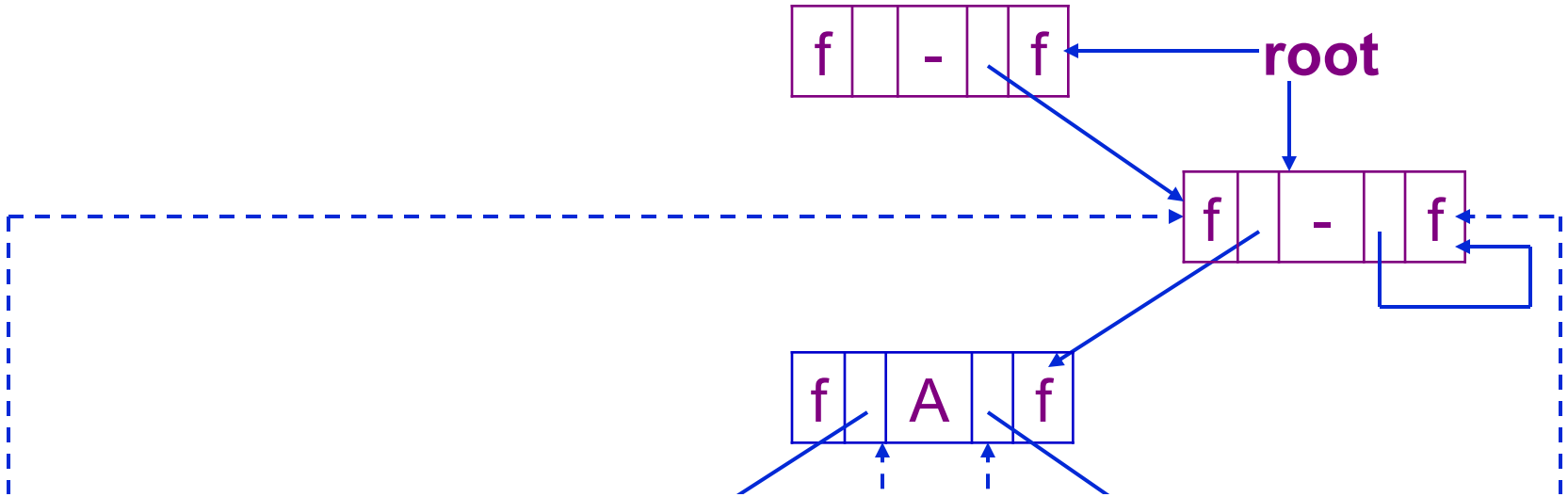
<
>

*

*

Let ThreadedInorderIterator be a nested class of ThreadedTree:

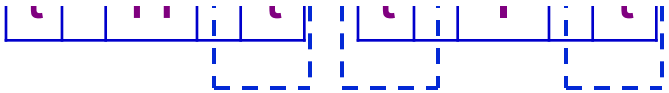
An empty threaded binary tree



we can see:

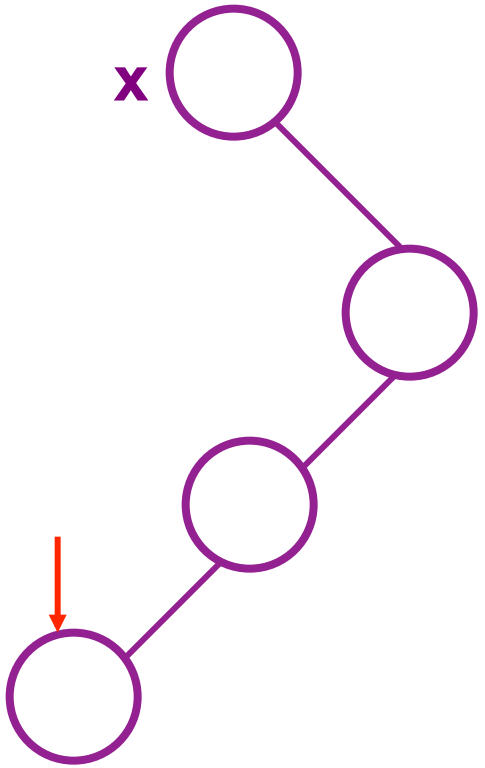
(1)The inorder successor of the head node is the first node in inorder;

(2)The inorder successor of the last node in inorder is the head node.



memory representation of threaded tree

Inorder Traversal of a Threaded Binary Tree


x

```
*                                   ::      ()
//

//
                        <  >*       =                 →
    (!                    →                )
            (!          →                  )
            =           →
                  =
    (                   ==          )
              0   //

              &                     →
```

< >

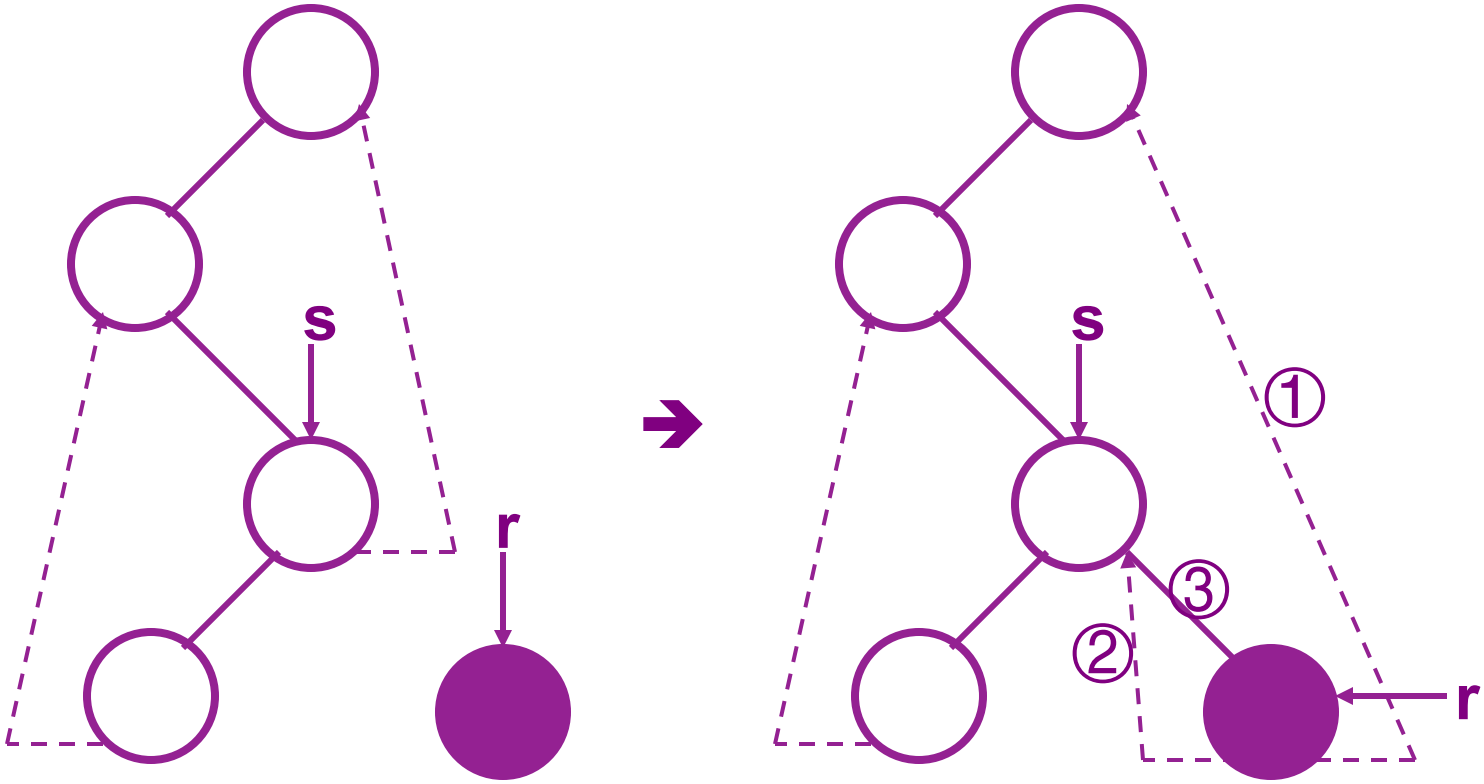
:: ()

(* = . () = . ())

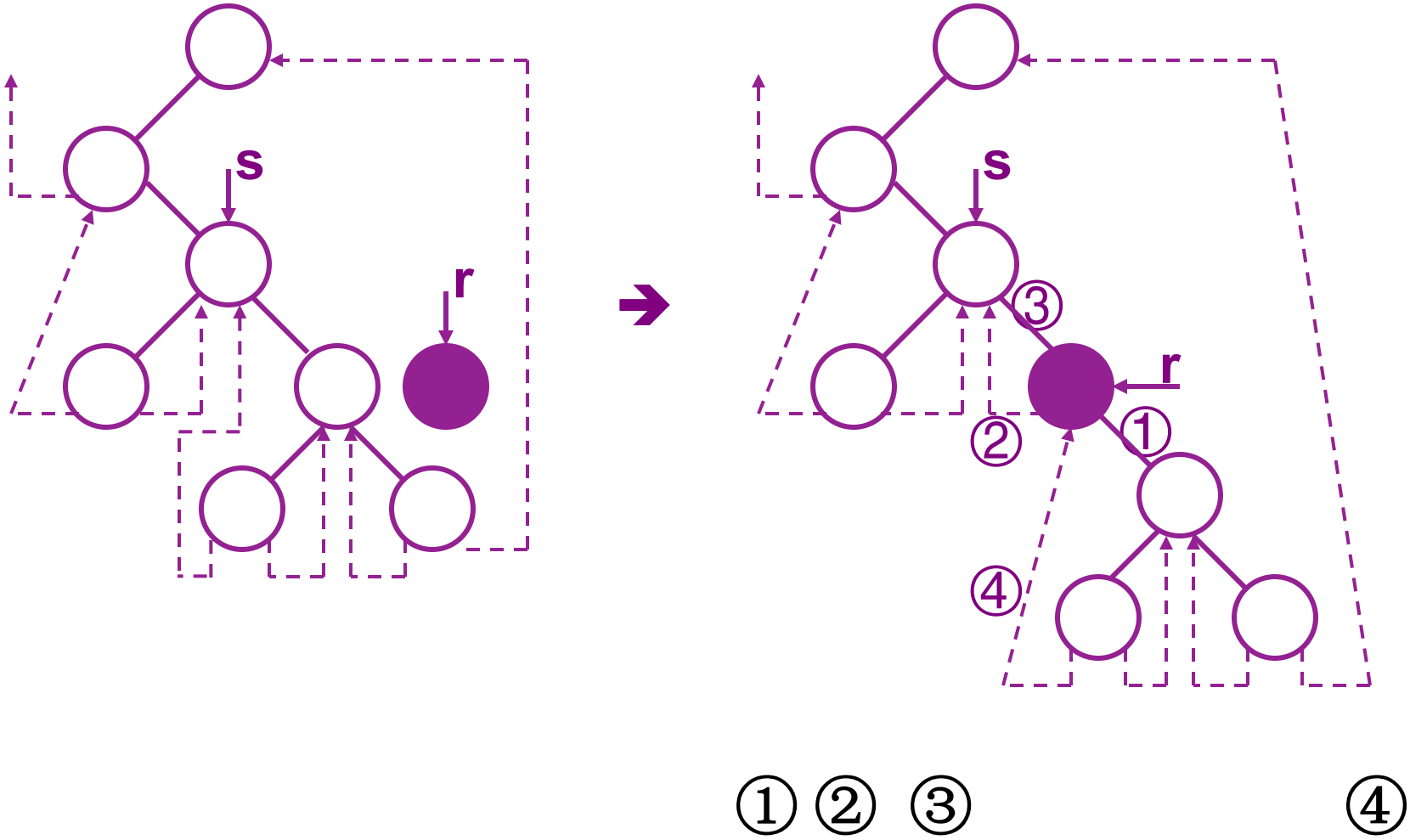
(*)

Inserting a Node into a Threaded Binary Tree

(1) If s rightThread==true, as:



(2) If s　rightThread==false, as:



①②　③　　④

< >

< >:: (< >* ,

< >*)

//

→ = → // ①

→ = → // ① != . ,

→ = // ②

→ // ②

→ = // ③

→ // ③

(! →) // (2)

< >* = ( ) // ④

→ = //

Exercises: P277-1, P278-4

,

.

ADT MaxHeap

< >

//

() = 0

//

& () = 0

//

(&) = 0

//

() = 0

//

.

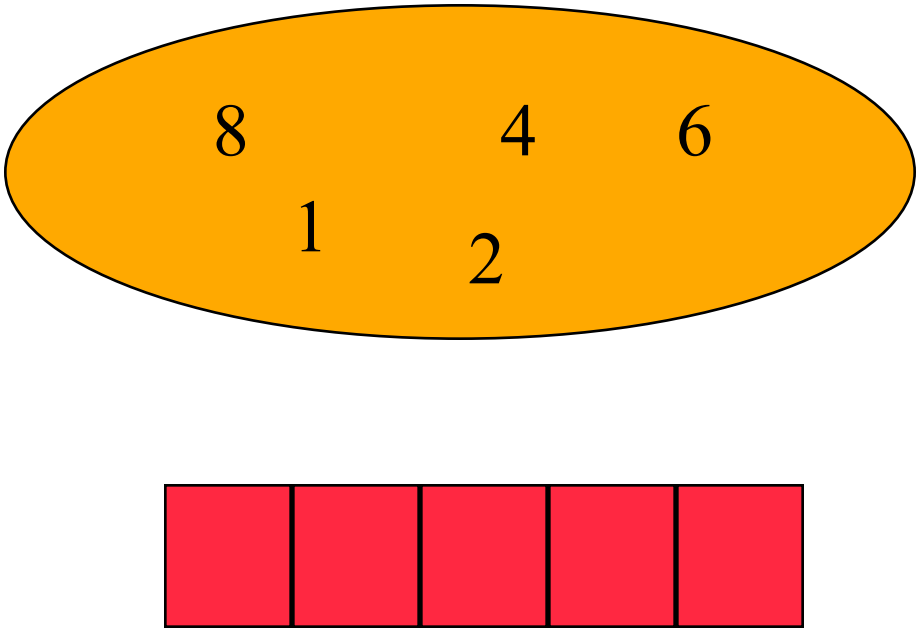
, , => (1)

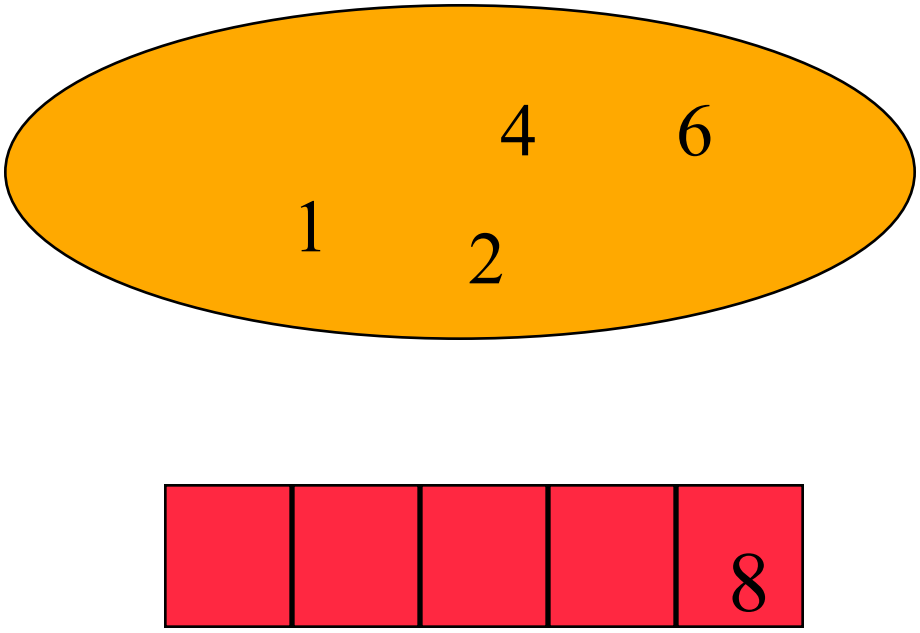
=> ( )

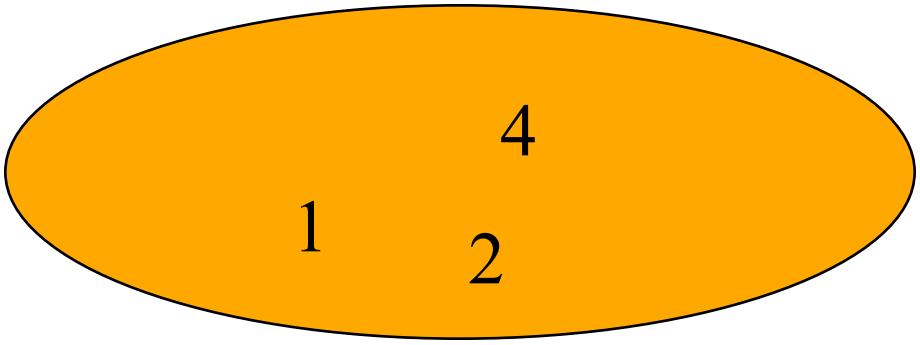
- ,
  ( )
- ,
  ( )

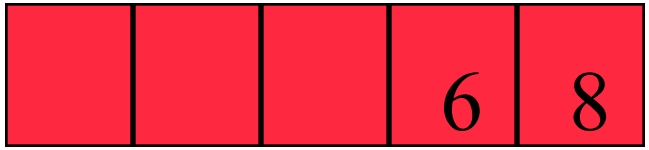
6, 8, 2, 4, 1

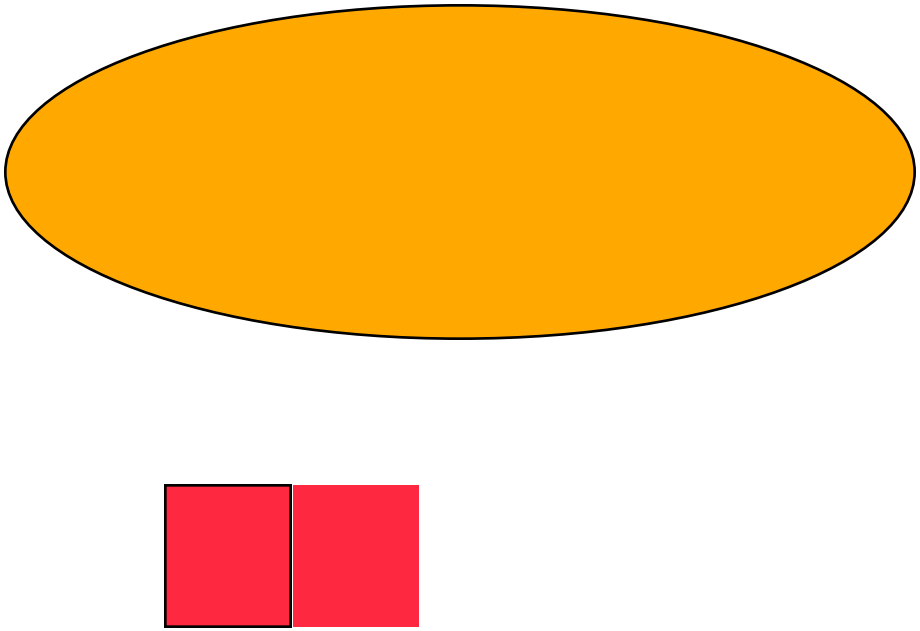
-
-

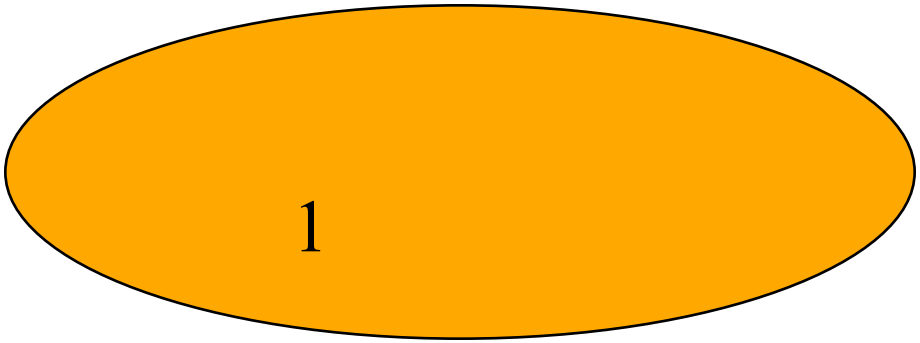

8
4
6
1
2

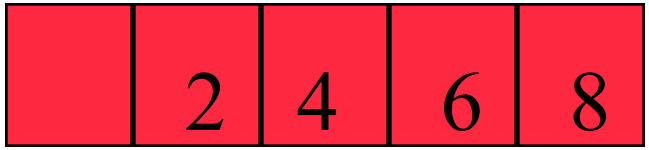

4
6
1
2
8

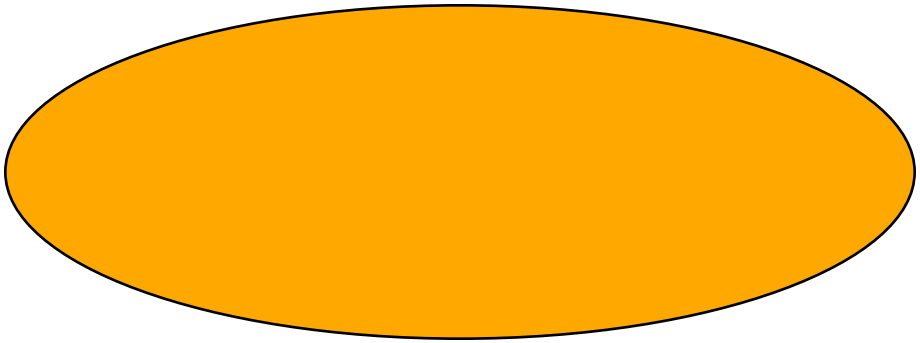

4
1
2

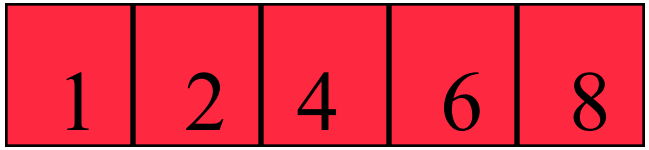

6
8




1


2
4
6
8




1
2
4
6
8

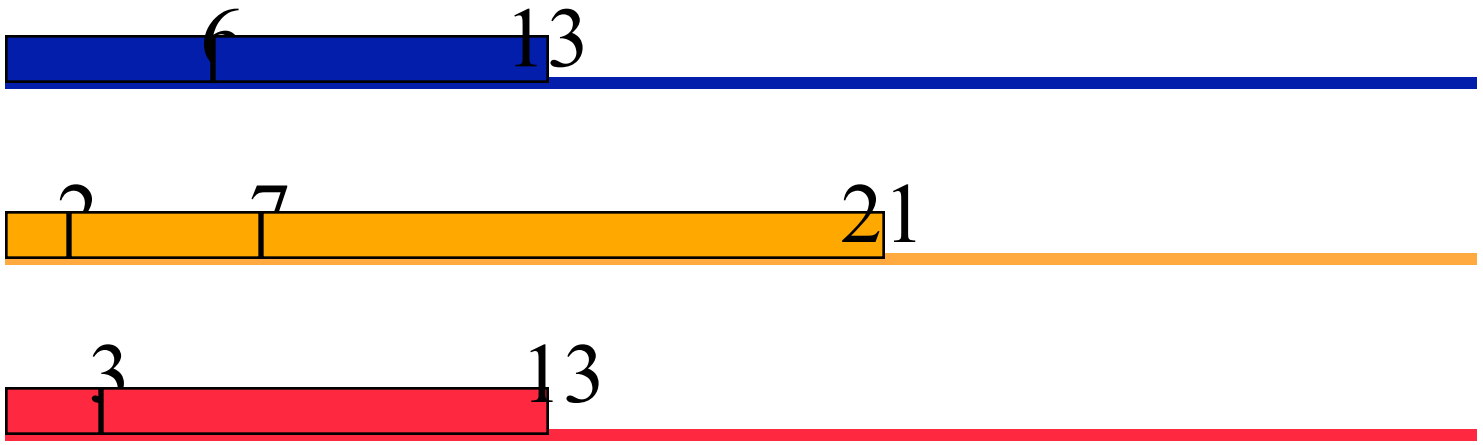
.

- => ( ) .
- => ( ) .
- ( ).
- (^{2})*

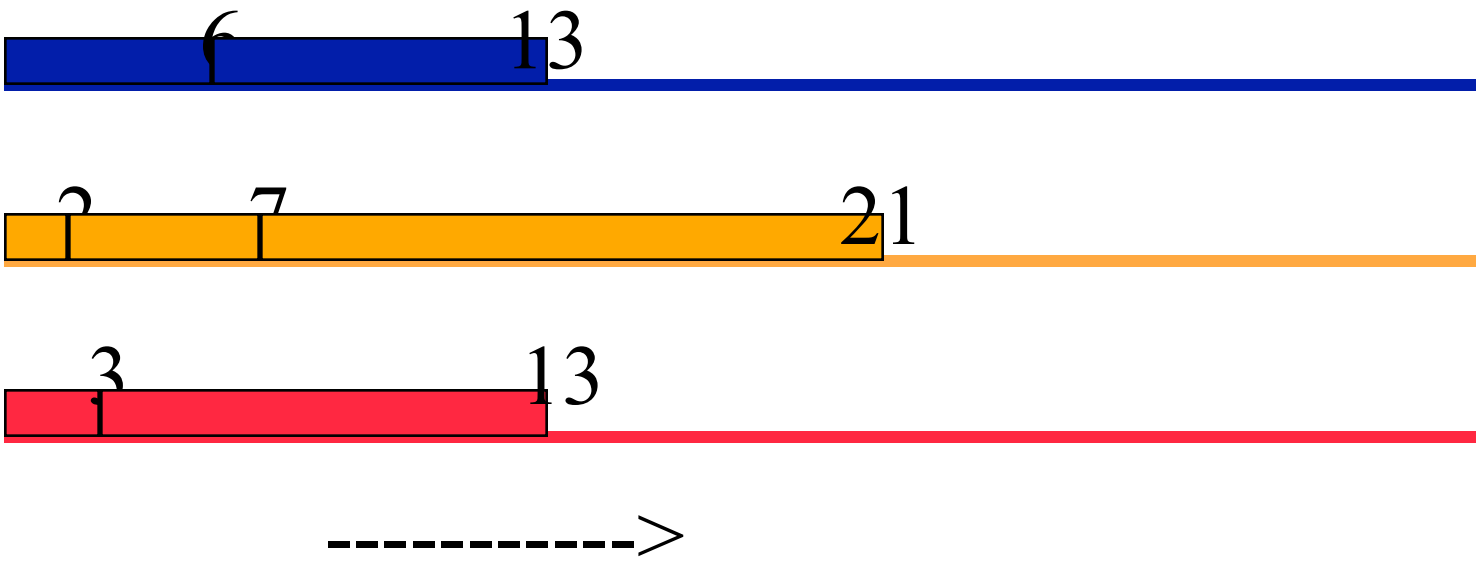
-
- /
-

3 7

6, 2, 3, 5, 10, 7, 14



---------->


6
13
2
7
21
3
13
---------->
= 21

14, 10, 7, 6, 5, 3, 2

.

- ( ) ( )

.

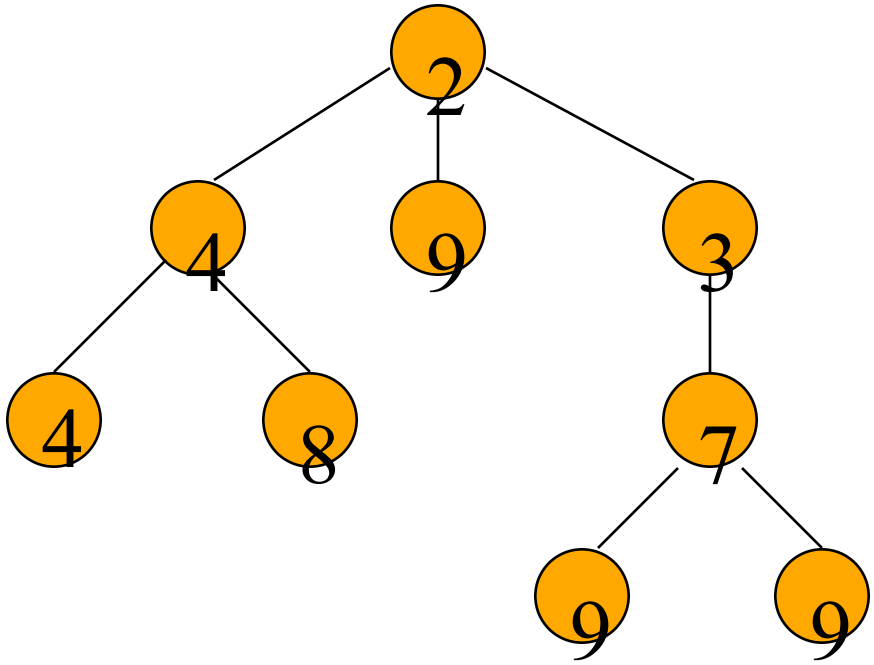
-

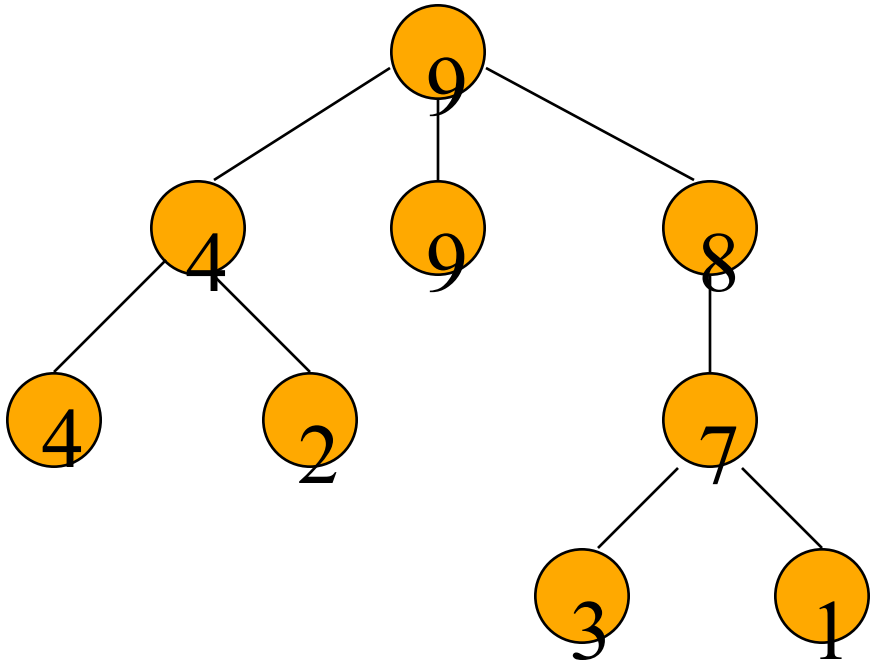
1 1

( )

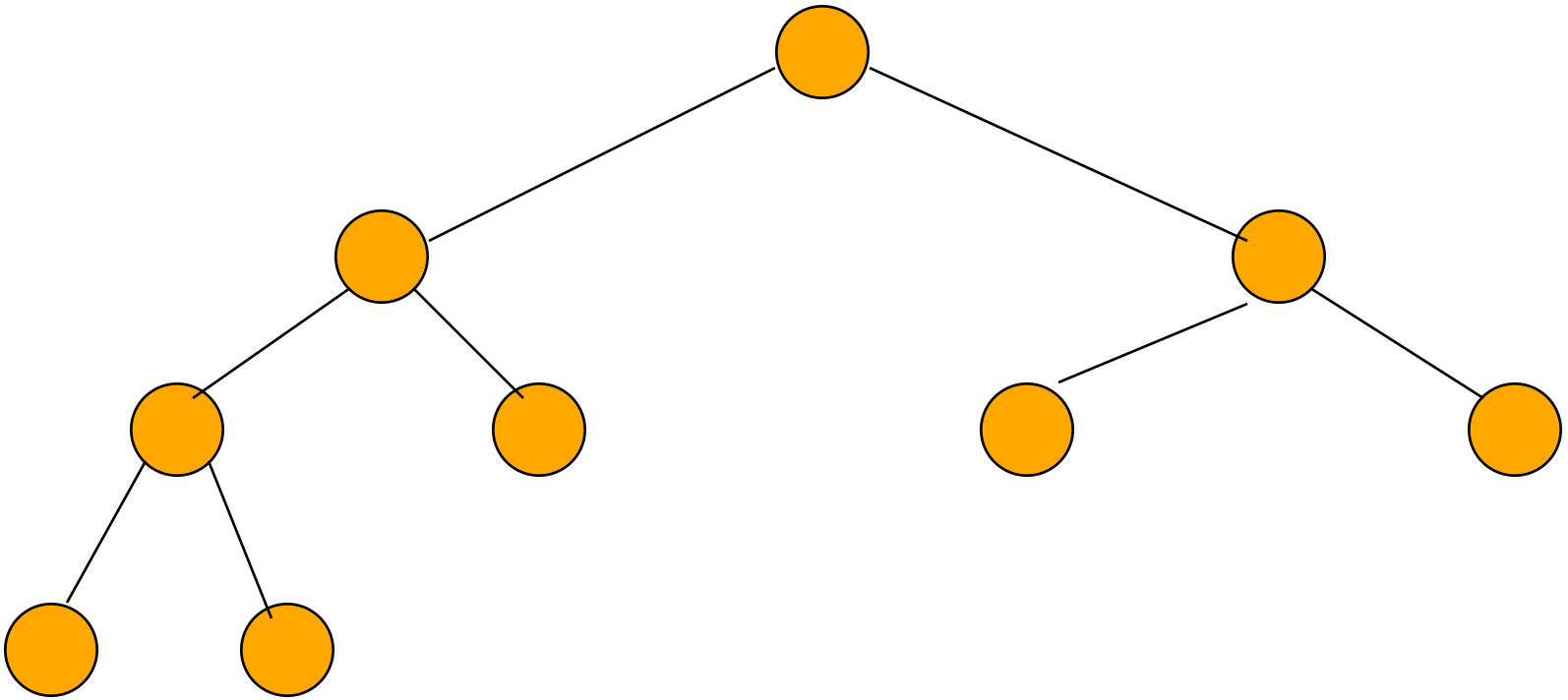
( )

(+) = (( ))


2
4
9
3
4
8
7
9
9

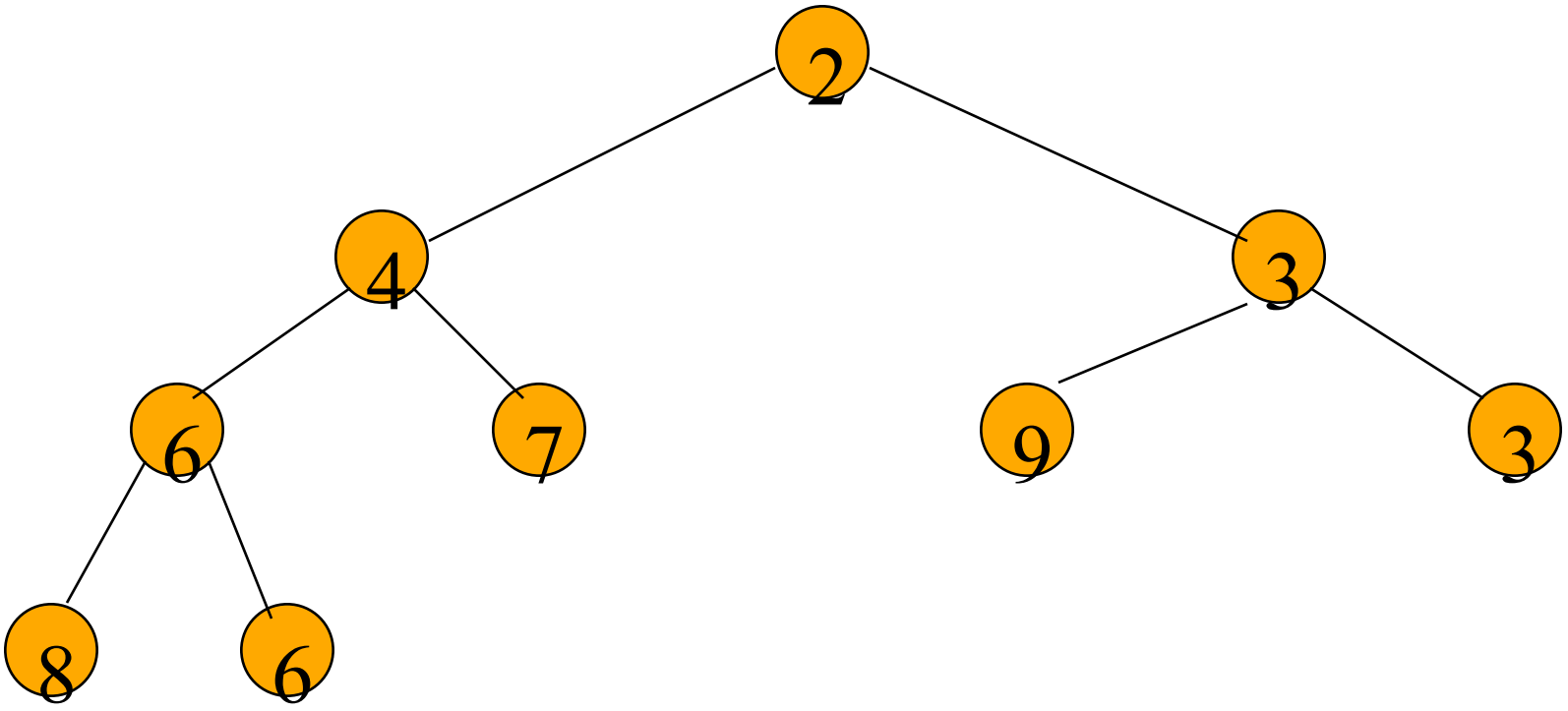

9
4
9
8
4
2
7
3
1

9



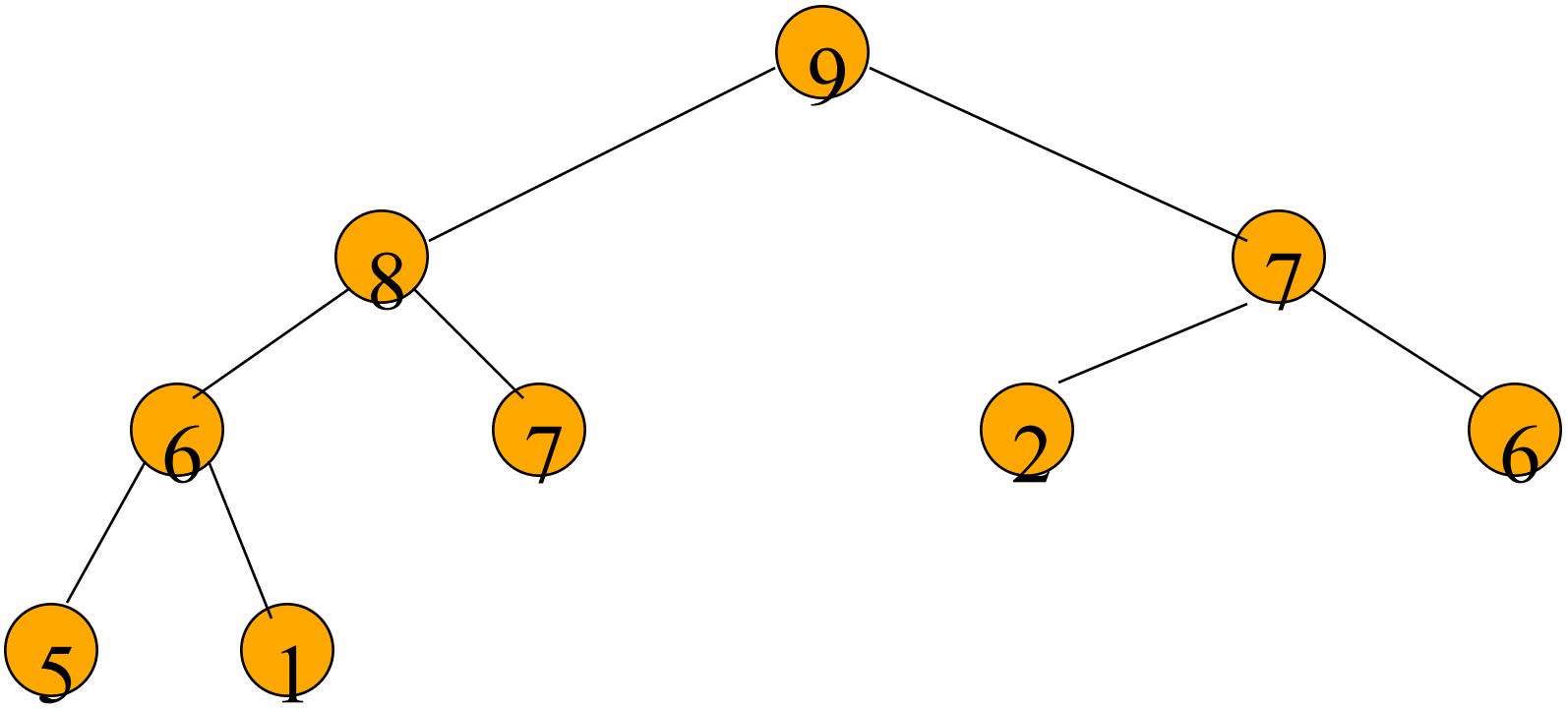
9

9



9

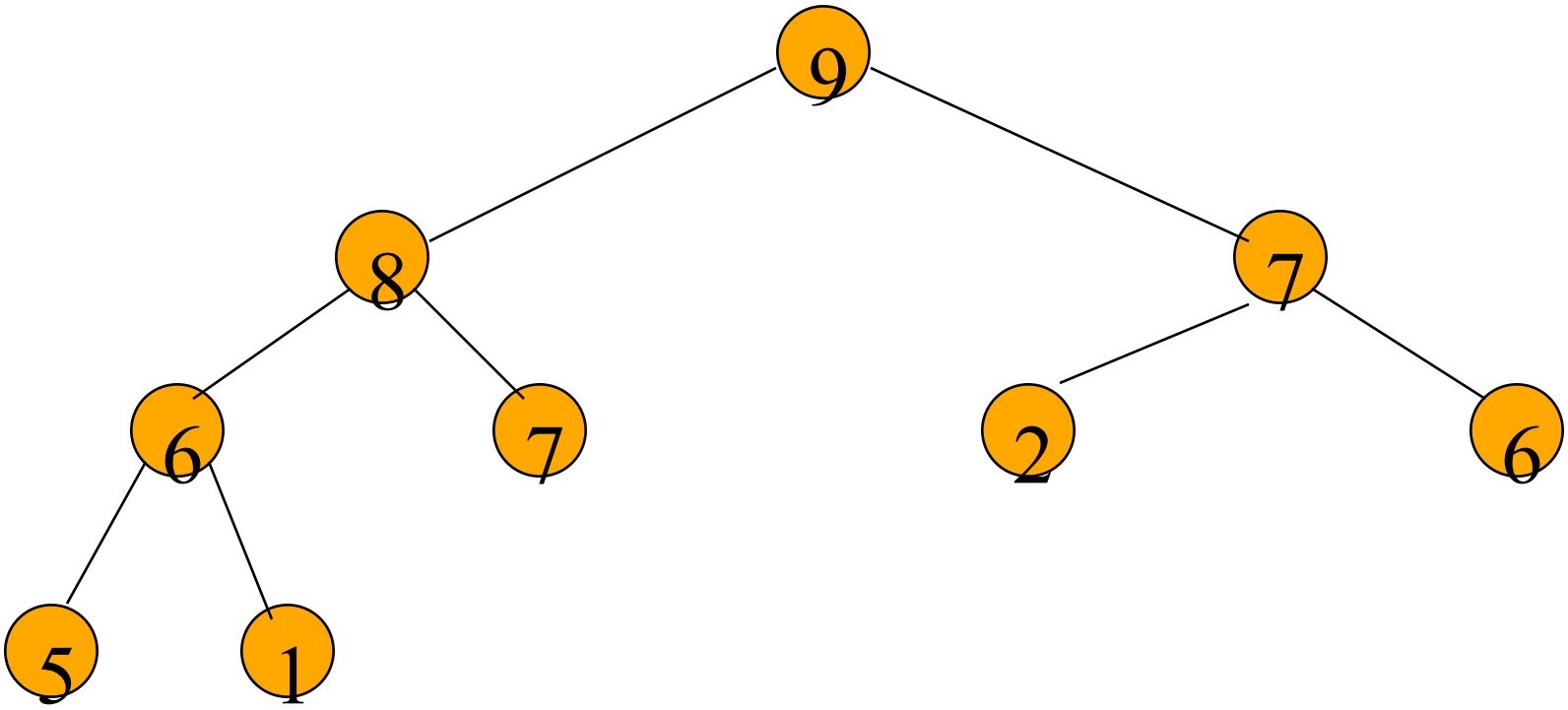
9

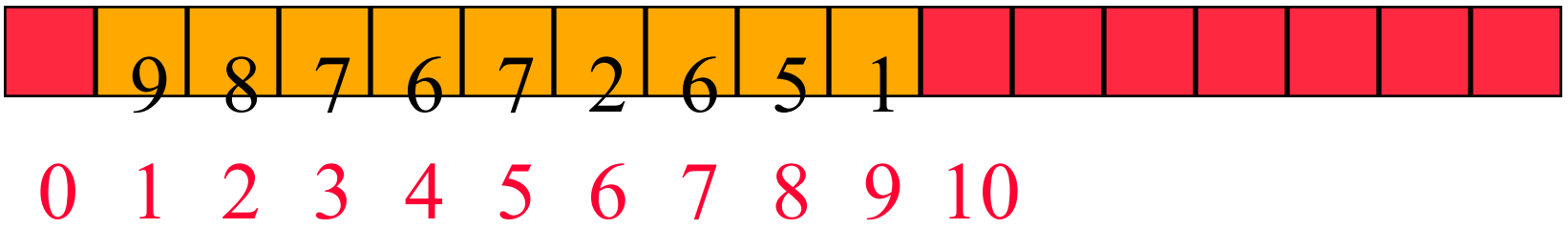


9

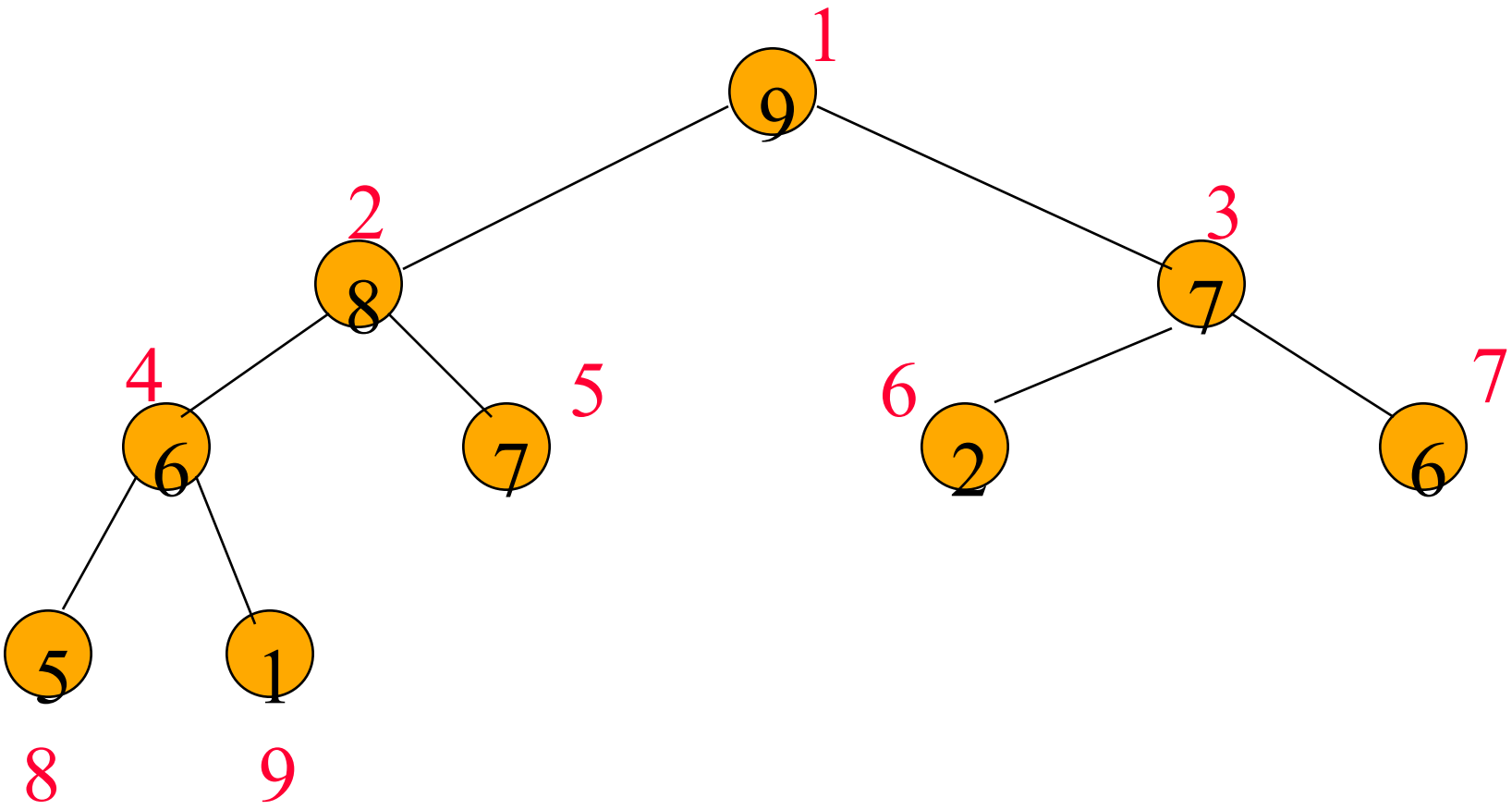
,

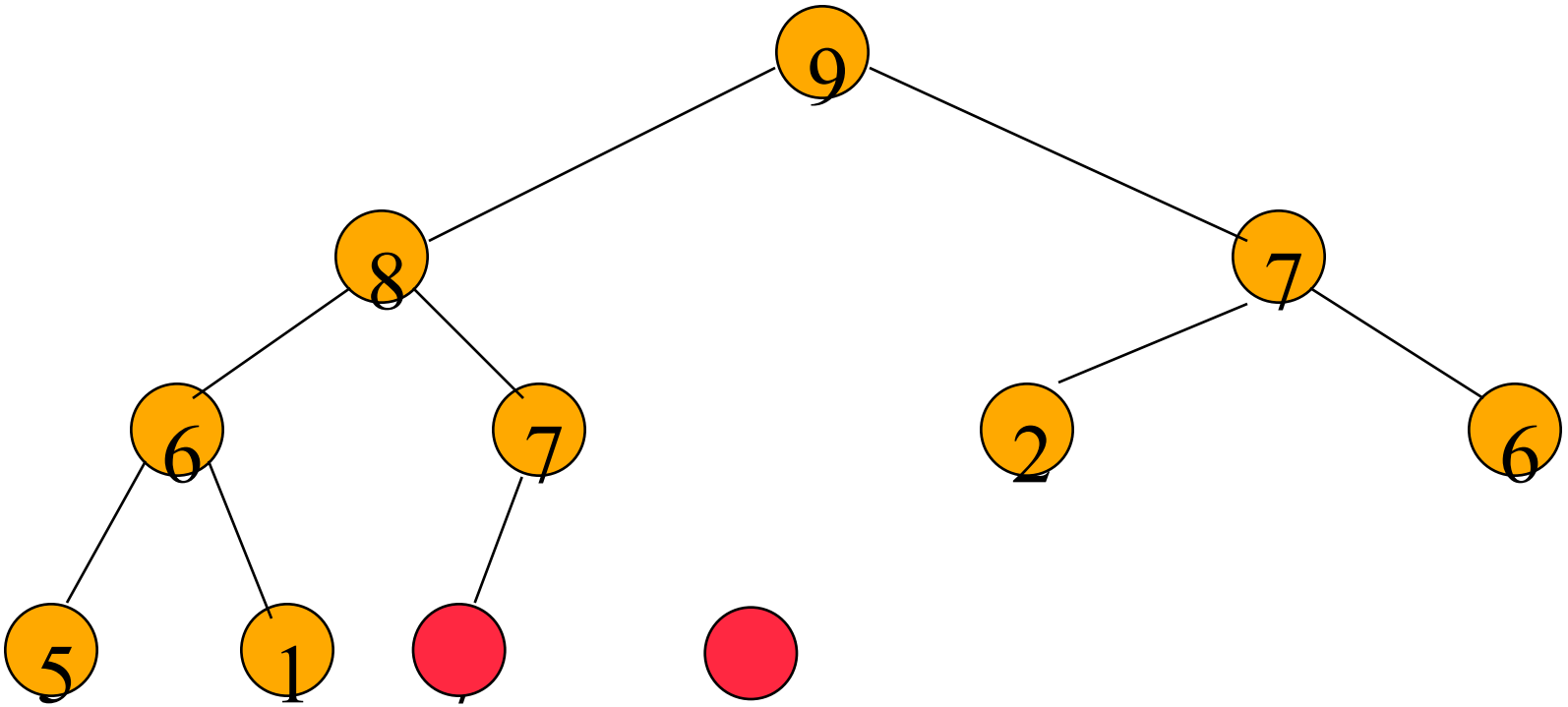
$_2$ (+1).


9
8
7
6
7
2
6
5
1

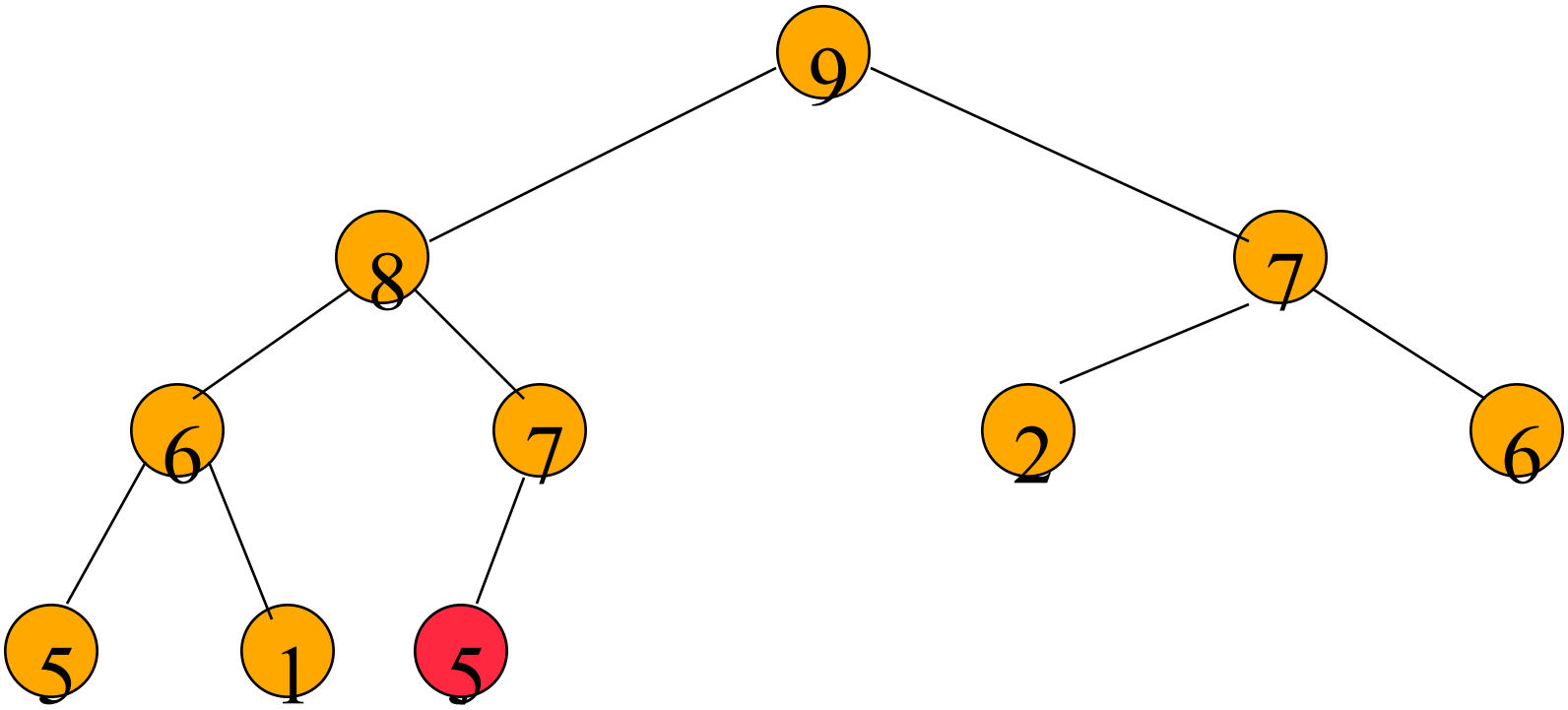

9 8 7 6 7 2 6 5 1
0 1 2 3 4 5 6 7 8 9 10

```
                <           >
                <  >::                  (                       =10)
    //
          (                   < 1)                                        >= 1
                    =
                     = 0
               =                          +1    //          0



                <           >
               &                  <  >::       ()

          (                ())
                             1
}
```

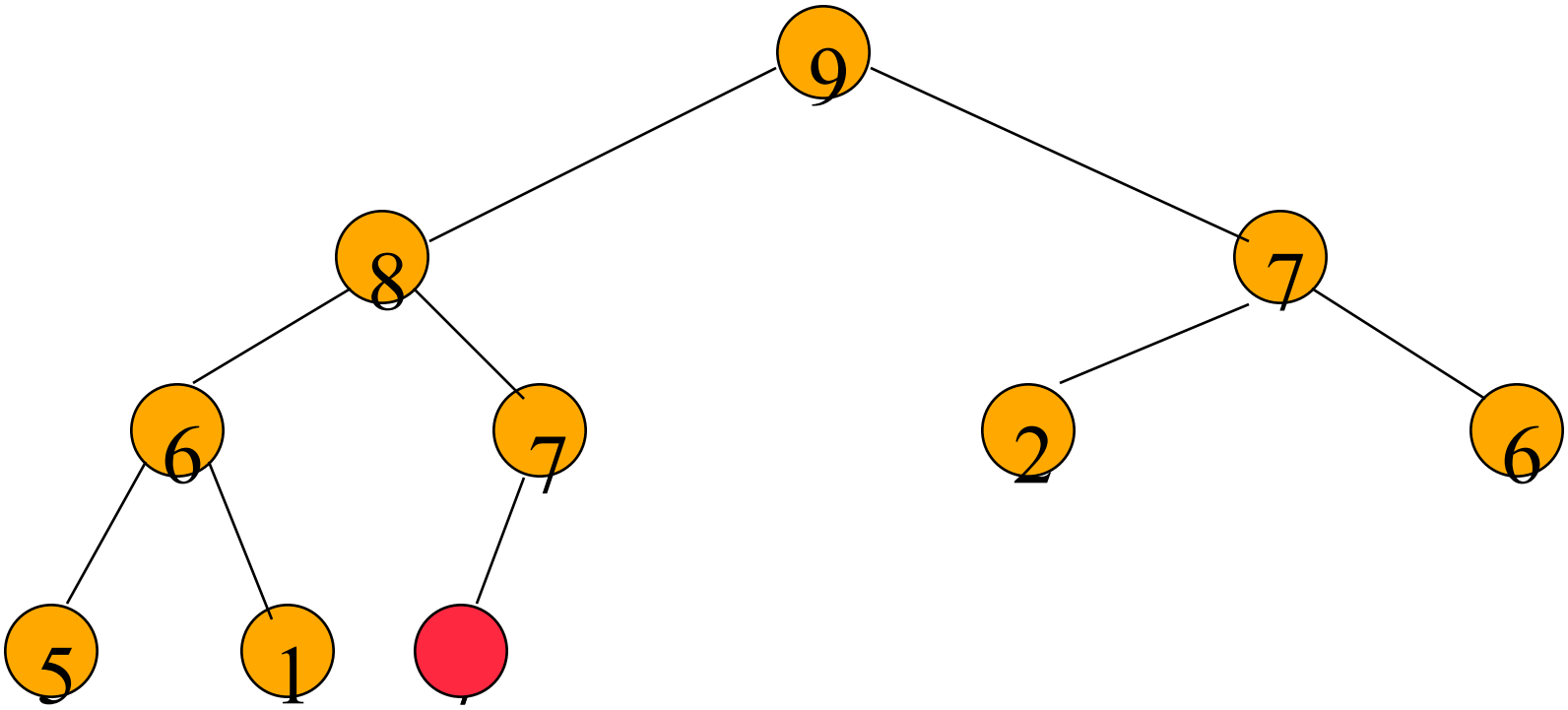

1
9
2
8
3
7
4
6
5
7
6
2
7
6
5
1
8
9

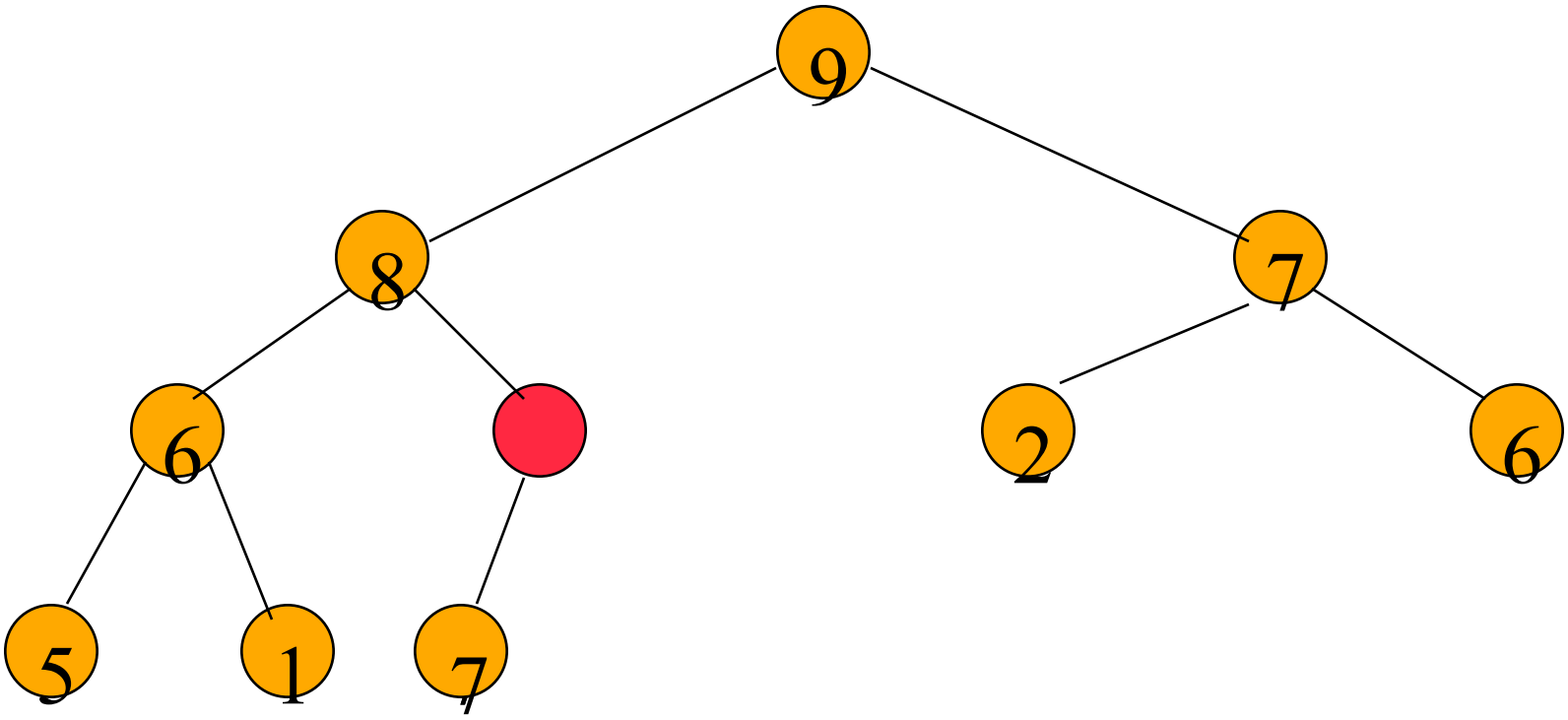


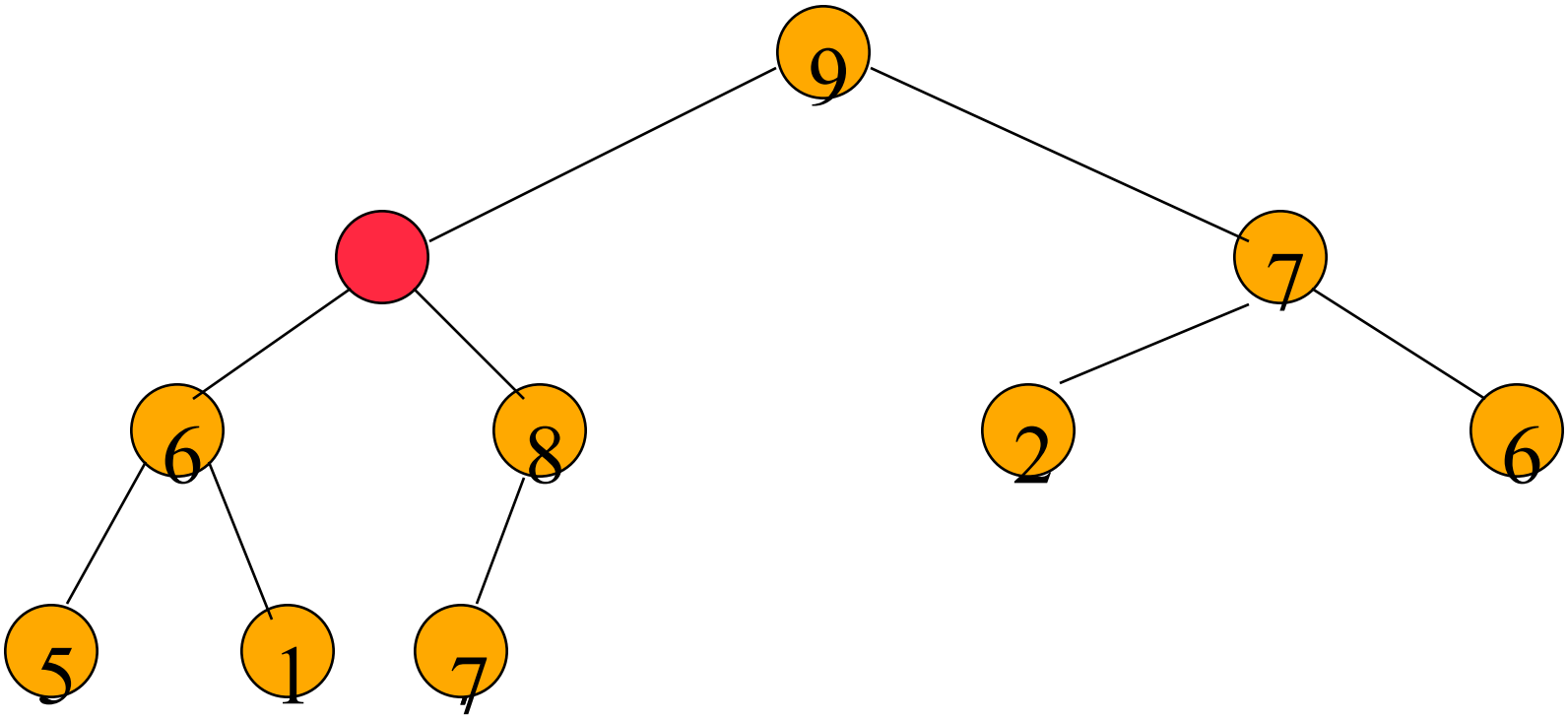
10 .

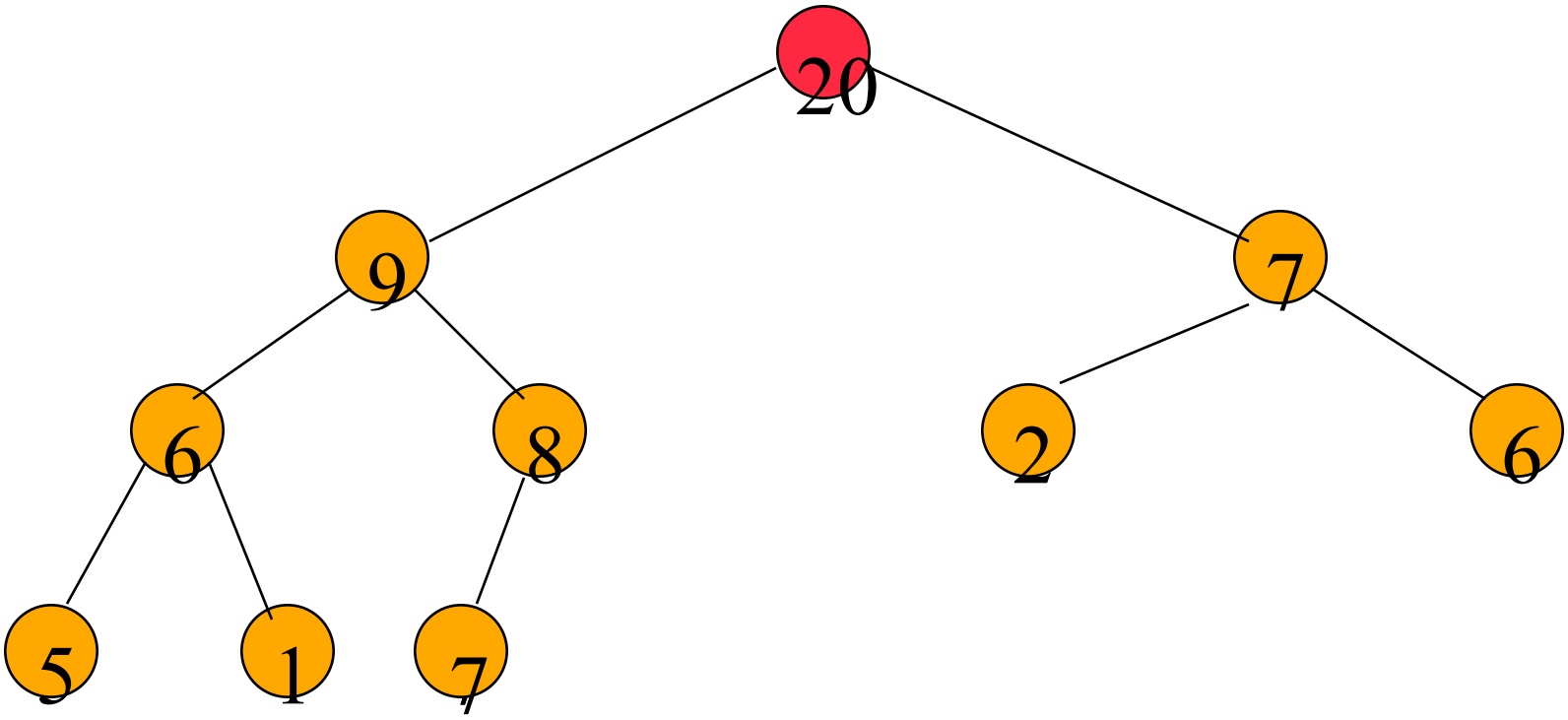


5.

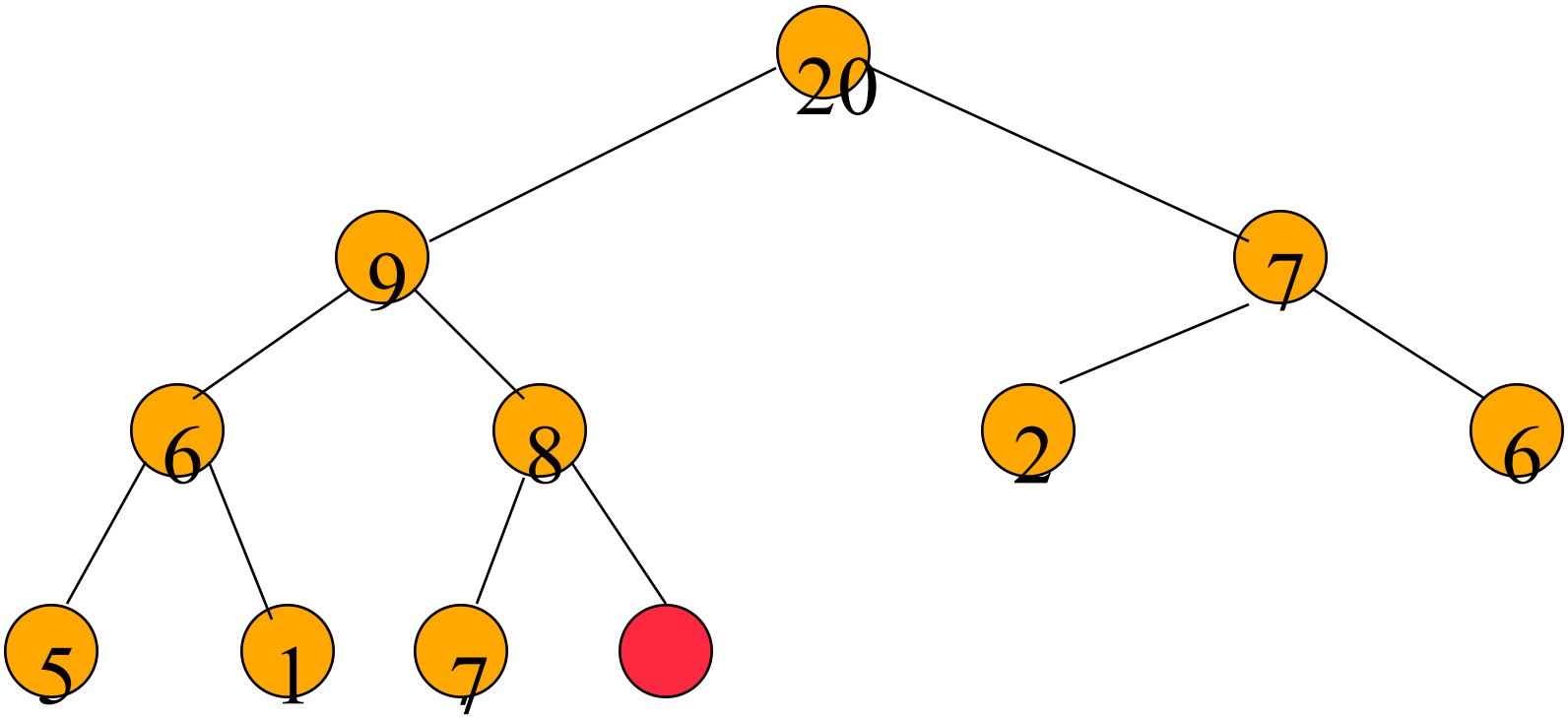

9
8
7
6
7
2
6
5
1


9
8
7
6
2
6
5
1
7

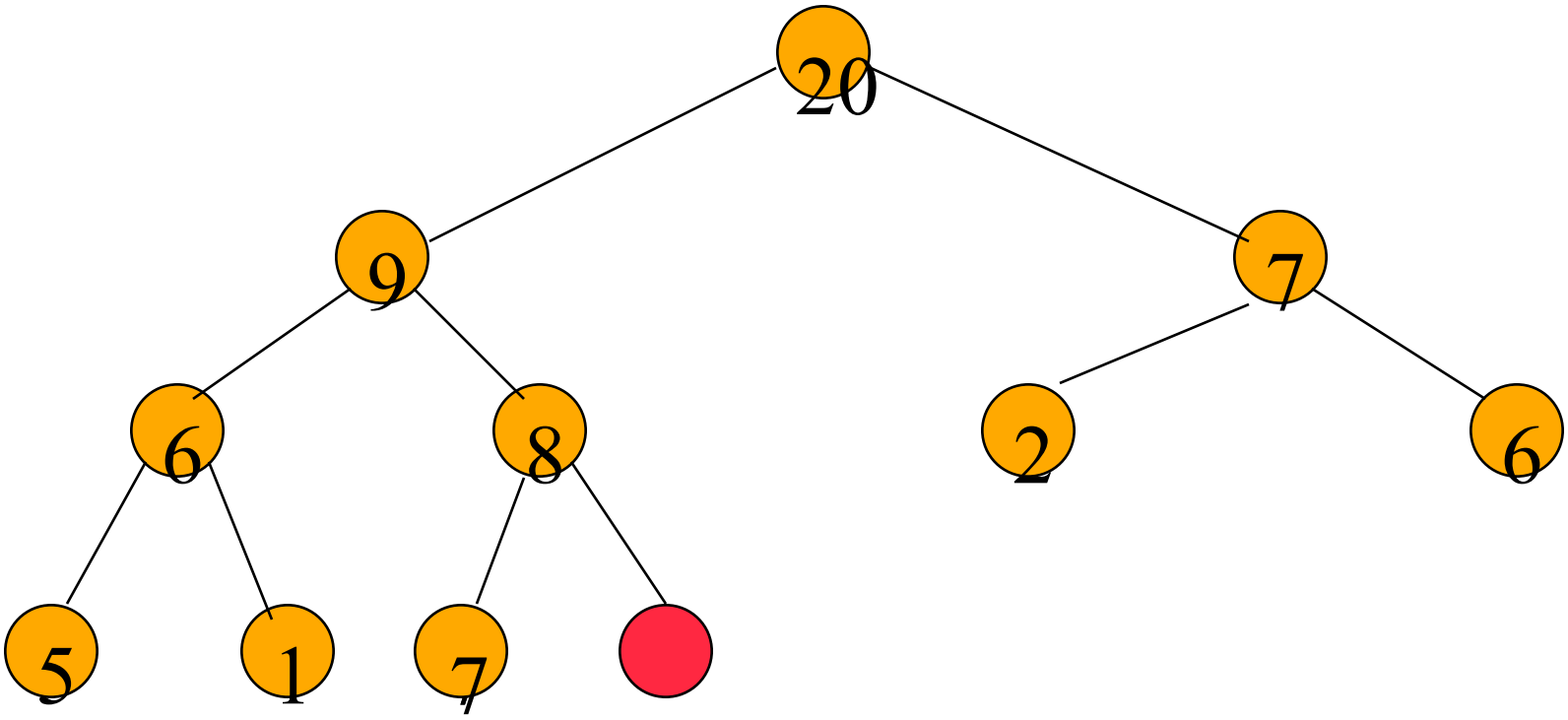

9
7
6
8
2
6
5
1
7



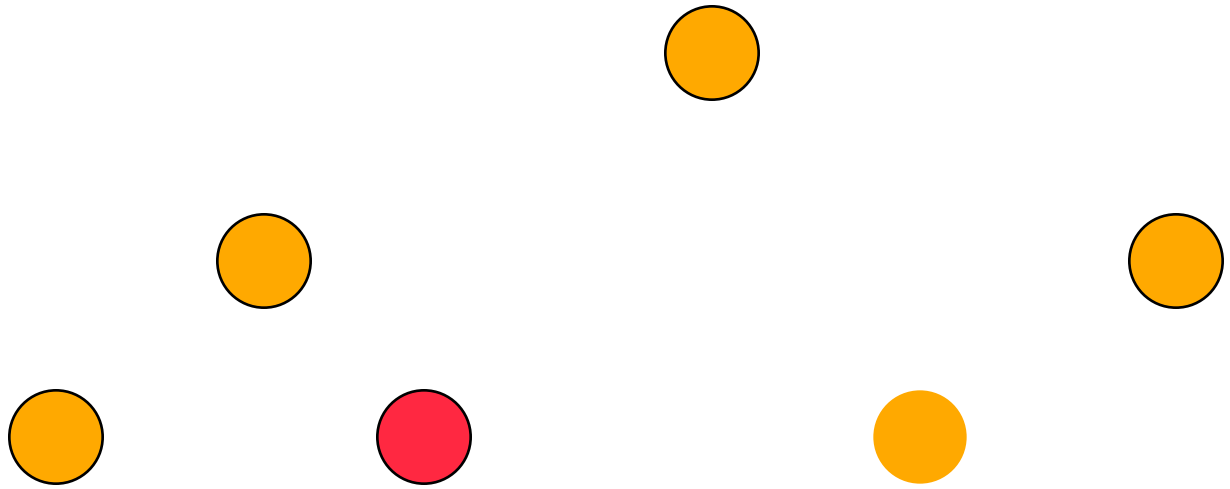
20.



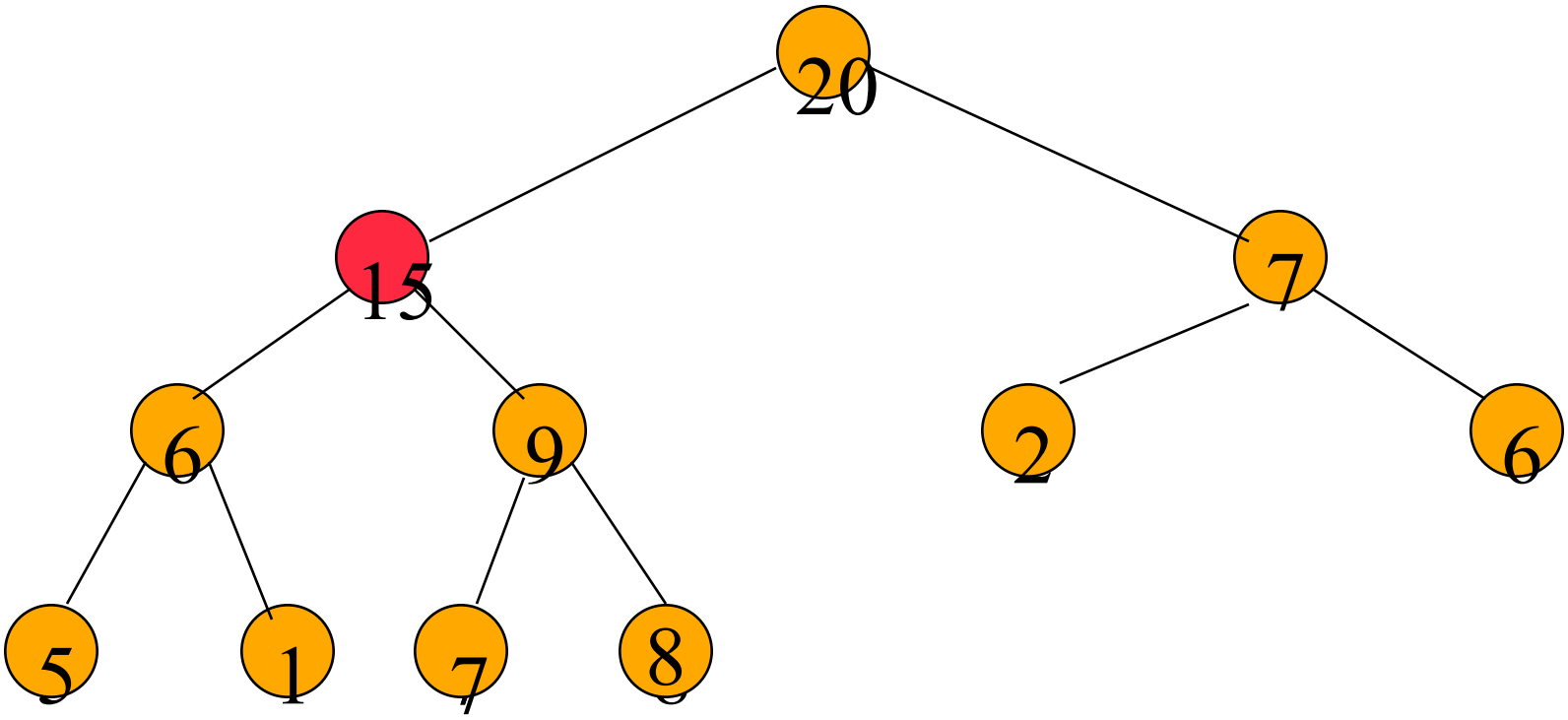
11 .



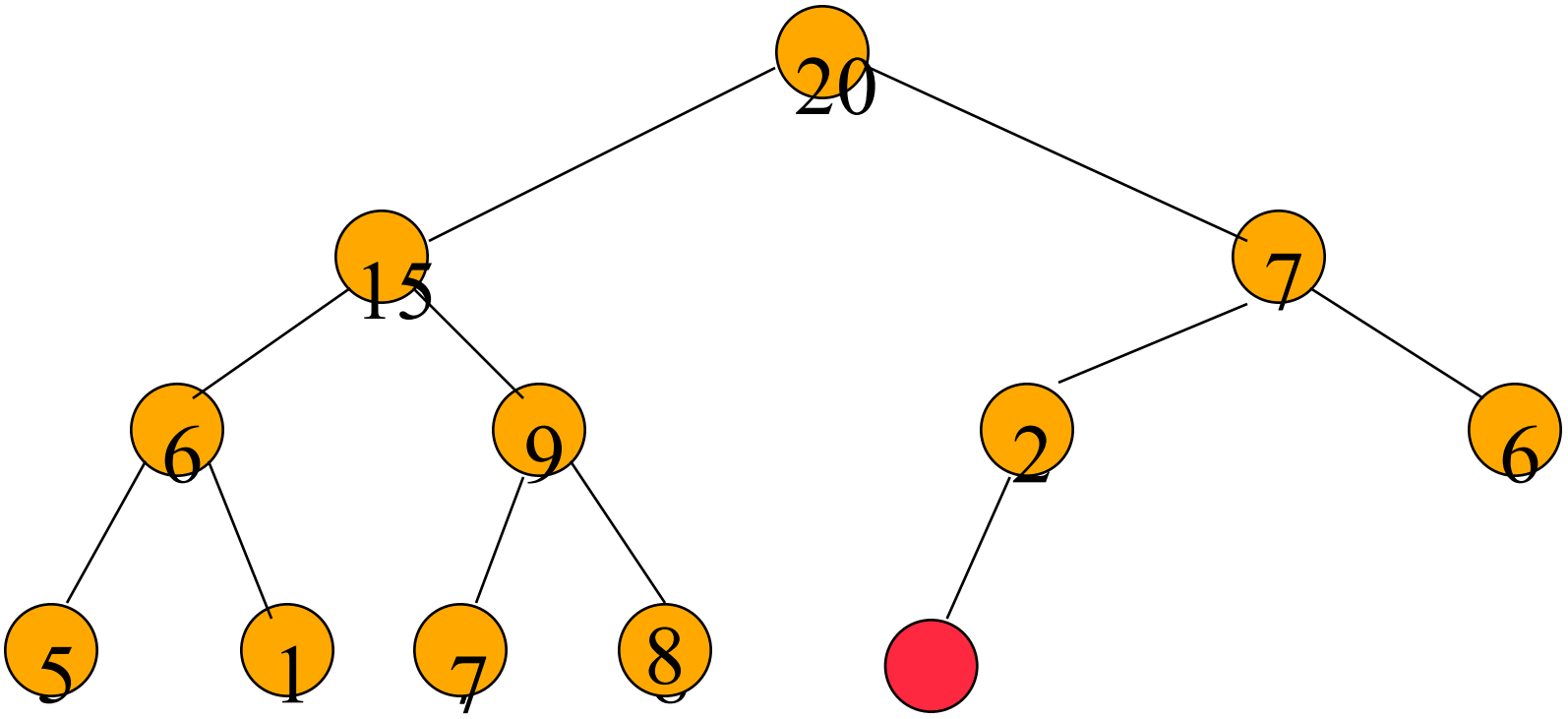
15.



15.



15.

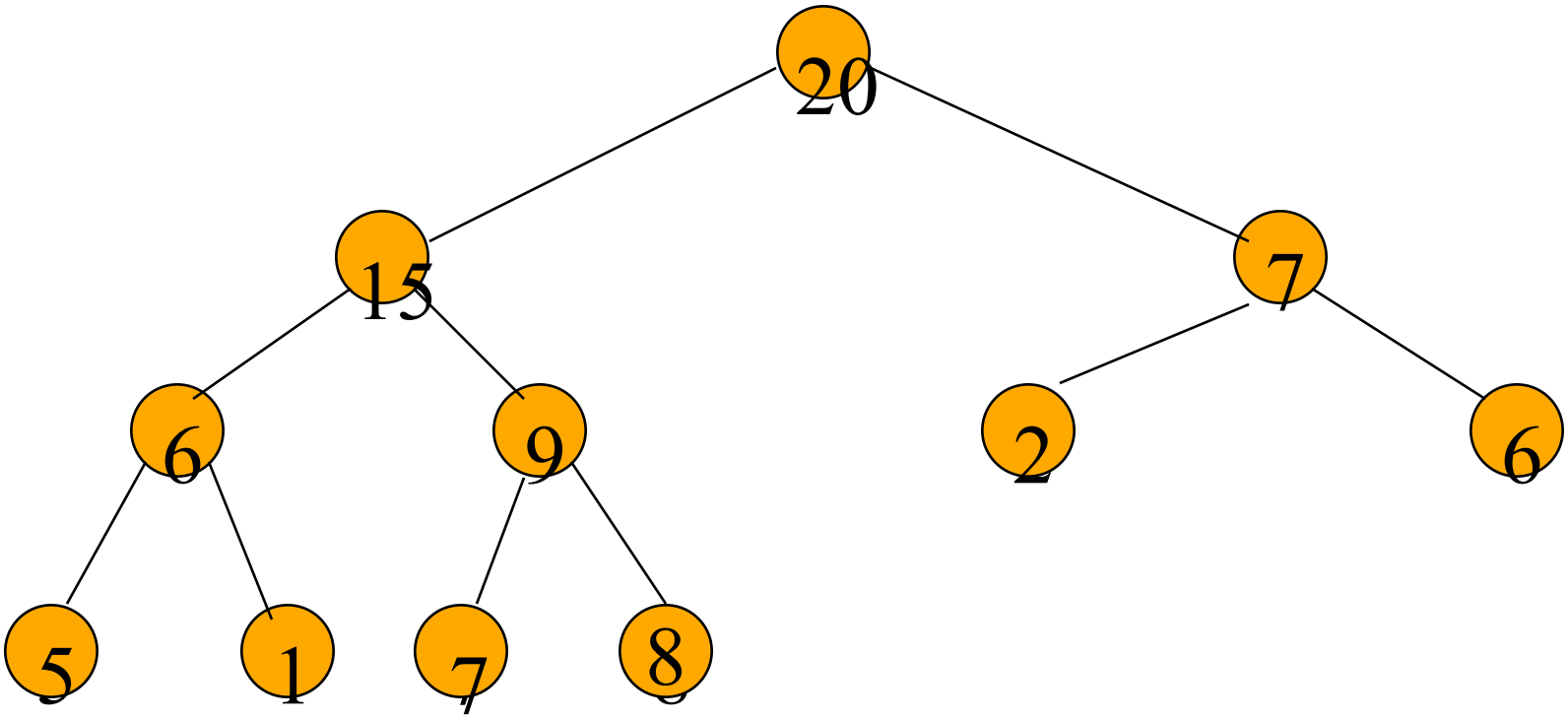

20
15
7
6
9
2
6
5
1
7
8

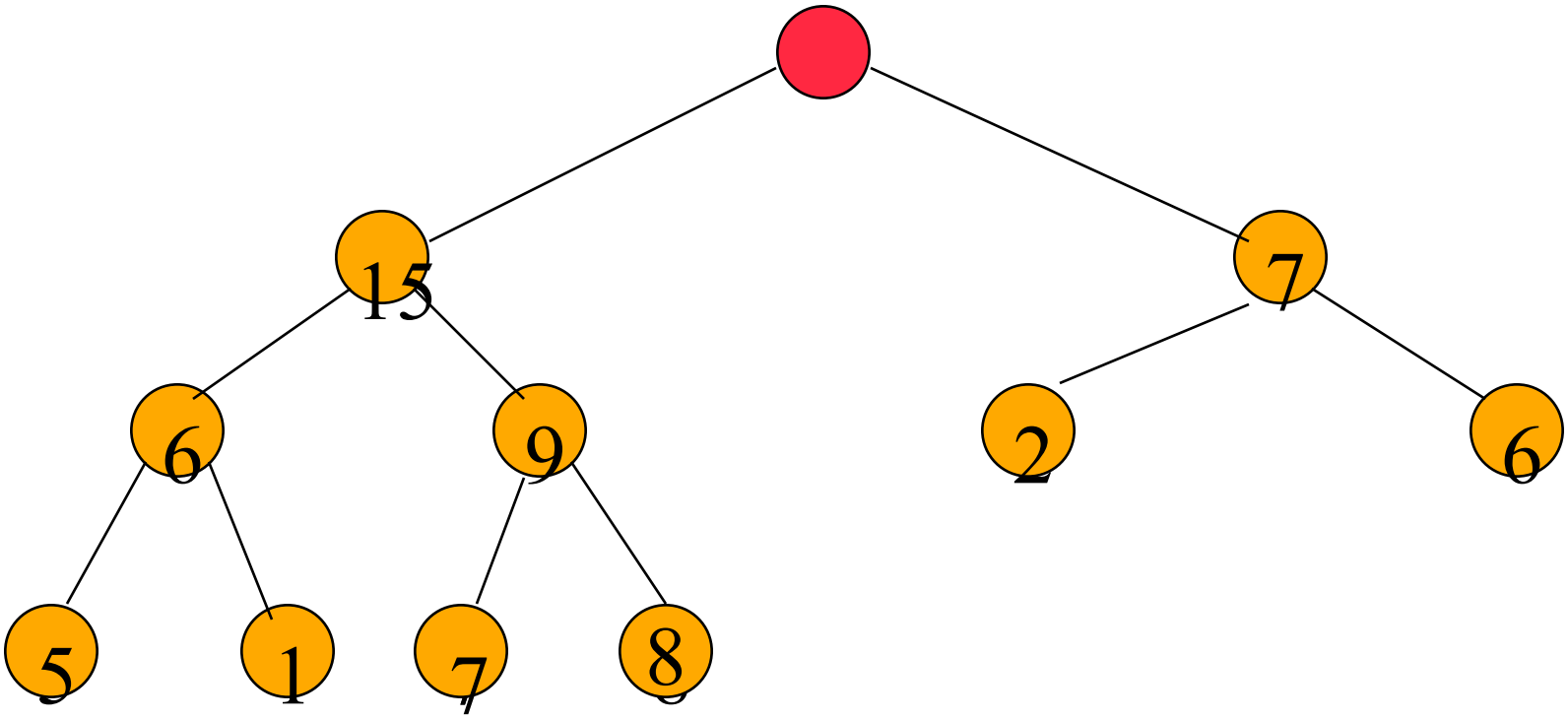
( ),

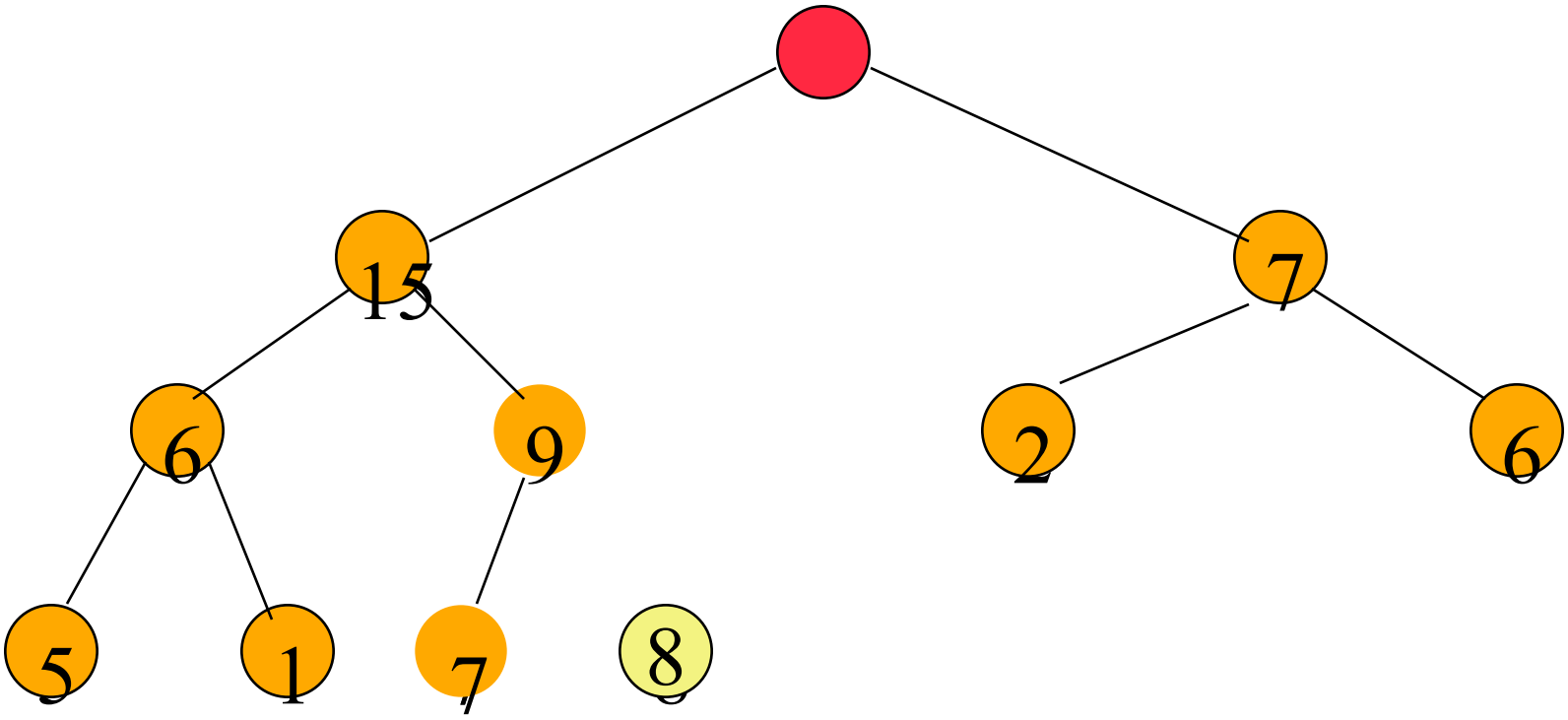
.

```
            <           >
                    <       >::       (             &   )
//
    (                ==              )  {  //
                       1   (       ,             , 2*             )
                    *= 2
```

= + 1: / 1.09 00 1 25.2 487.4 0 24 8.83

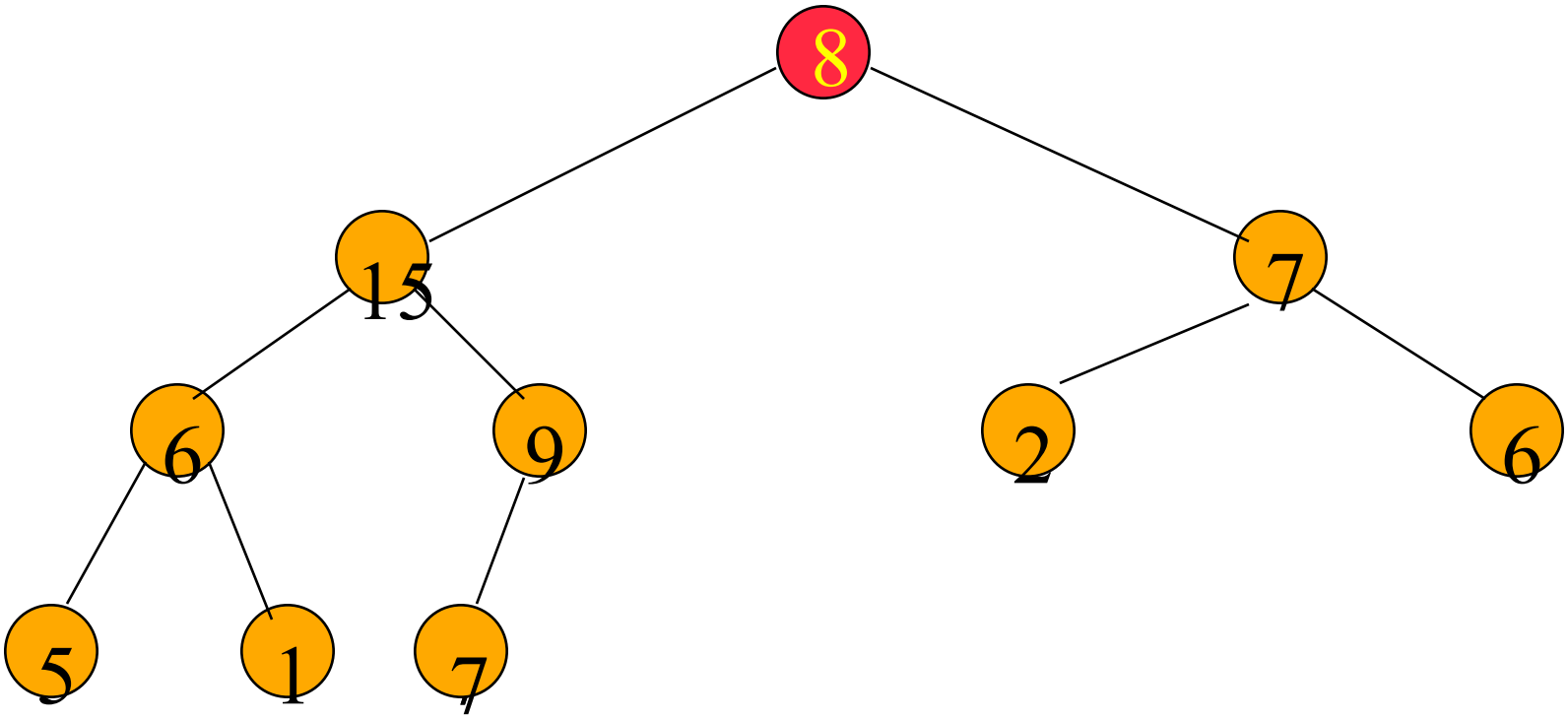

20
15
7
6
9
2
6
5
1
7
8


15
7
6
9
2
6
5
1
7
8

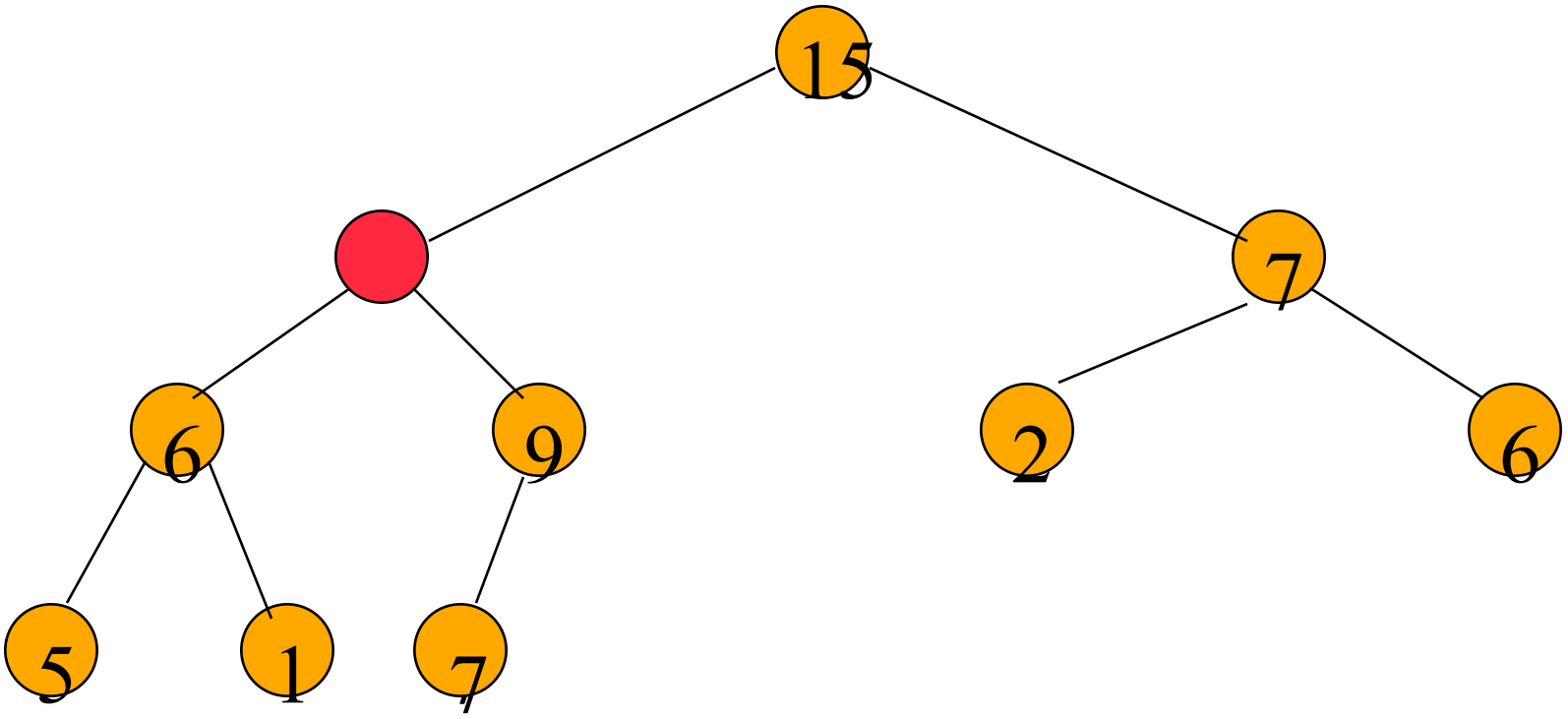


10

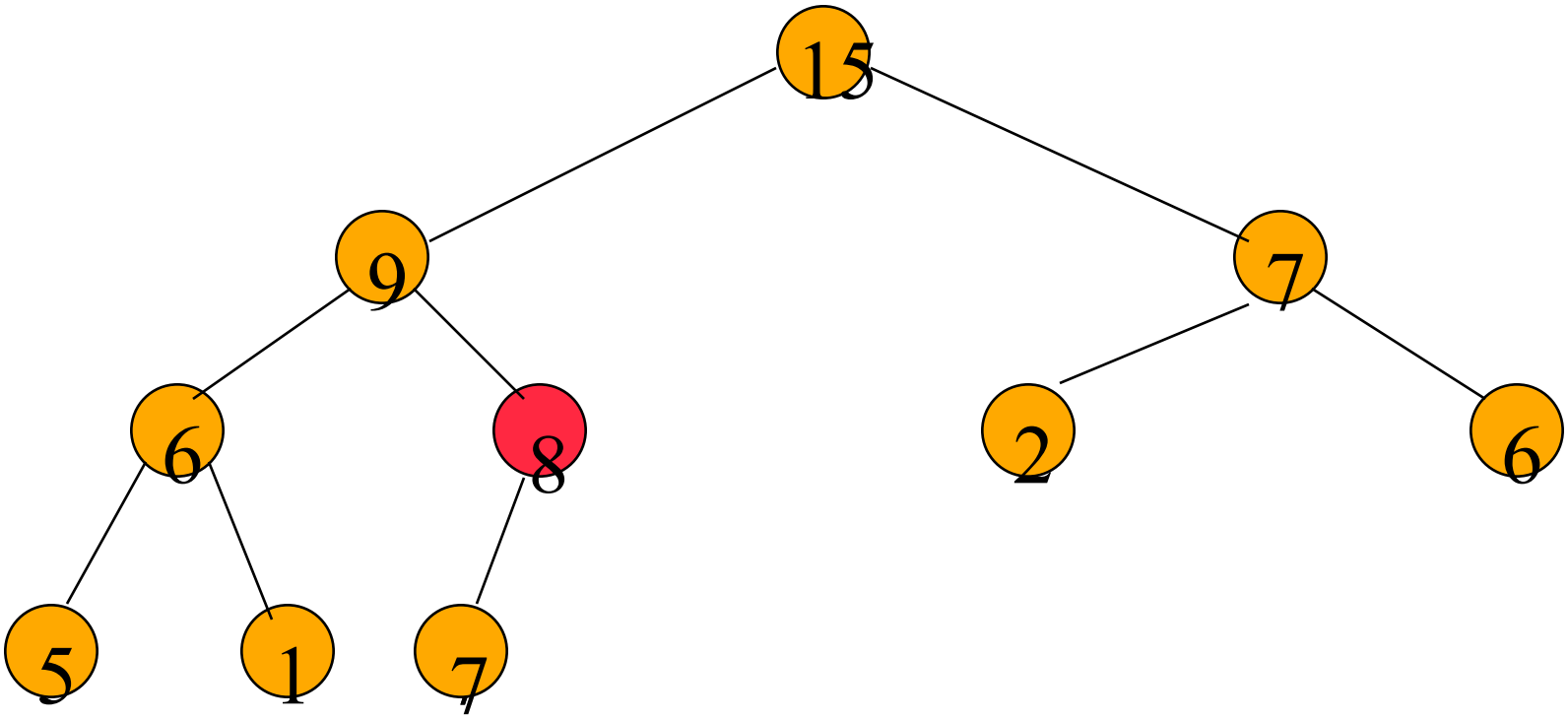
8



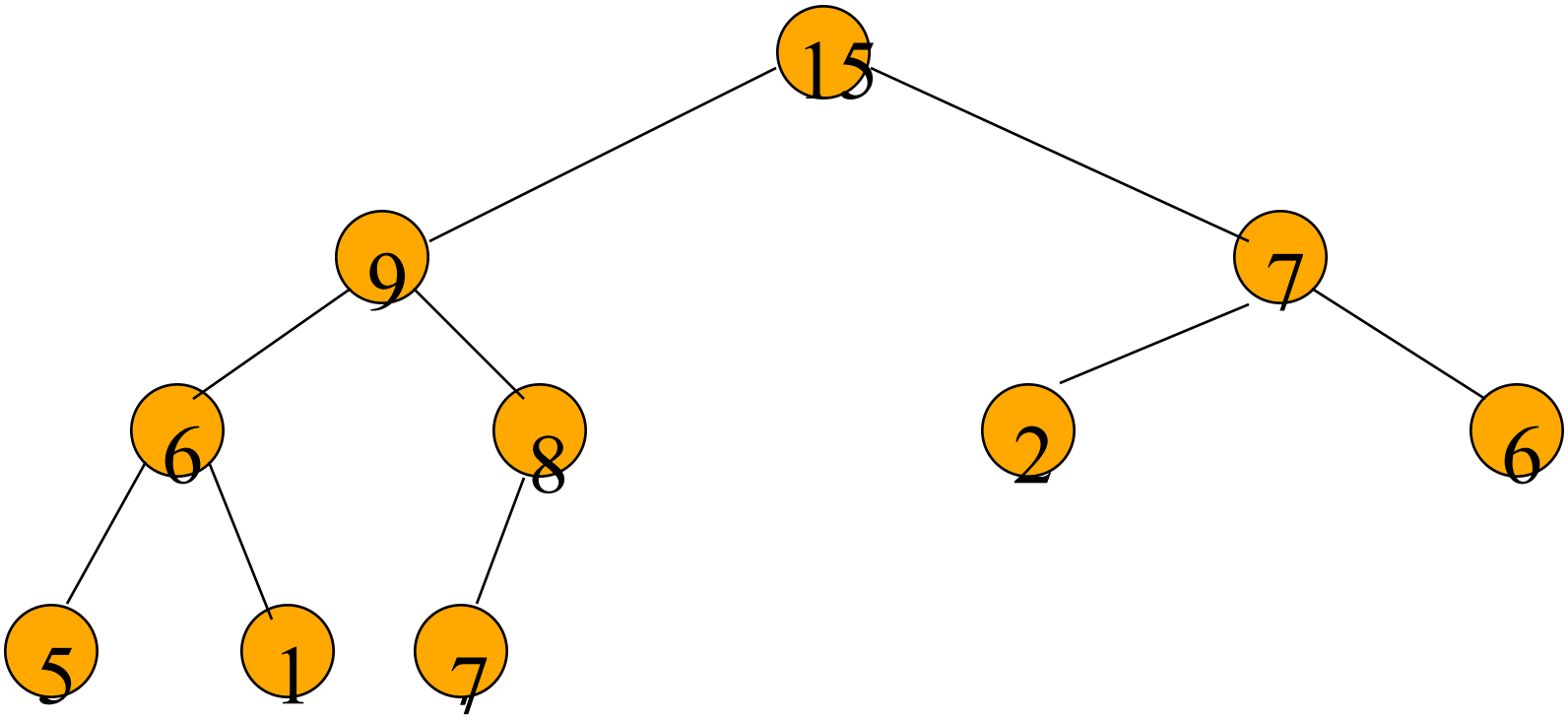
8 .



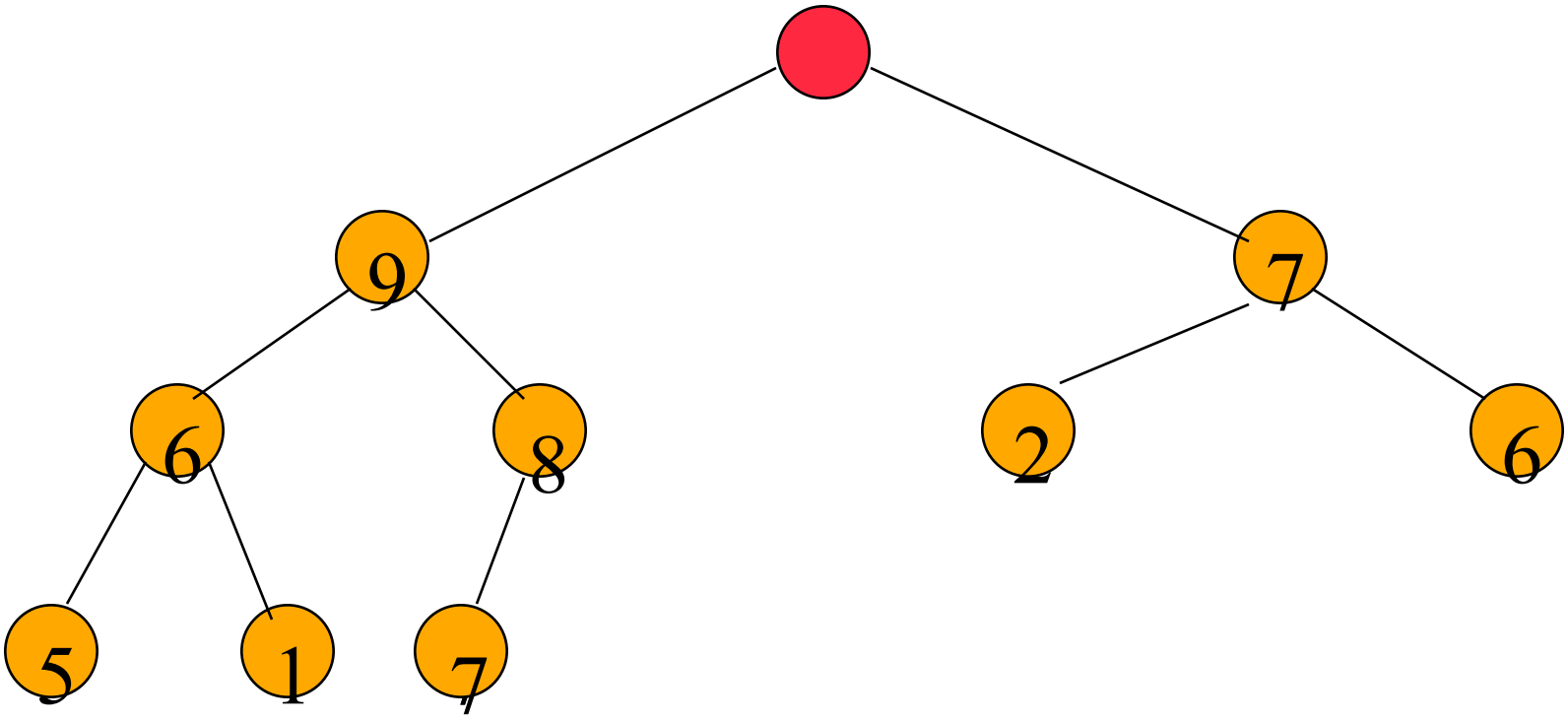
8

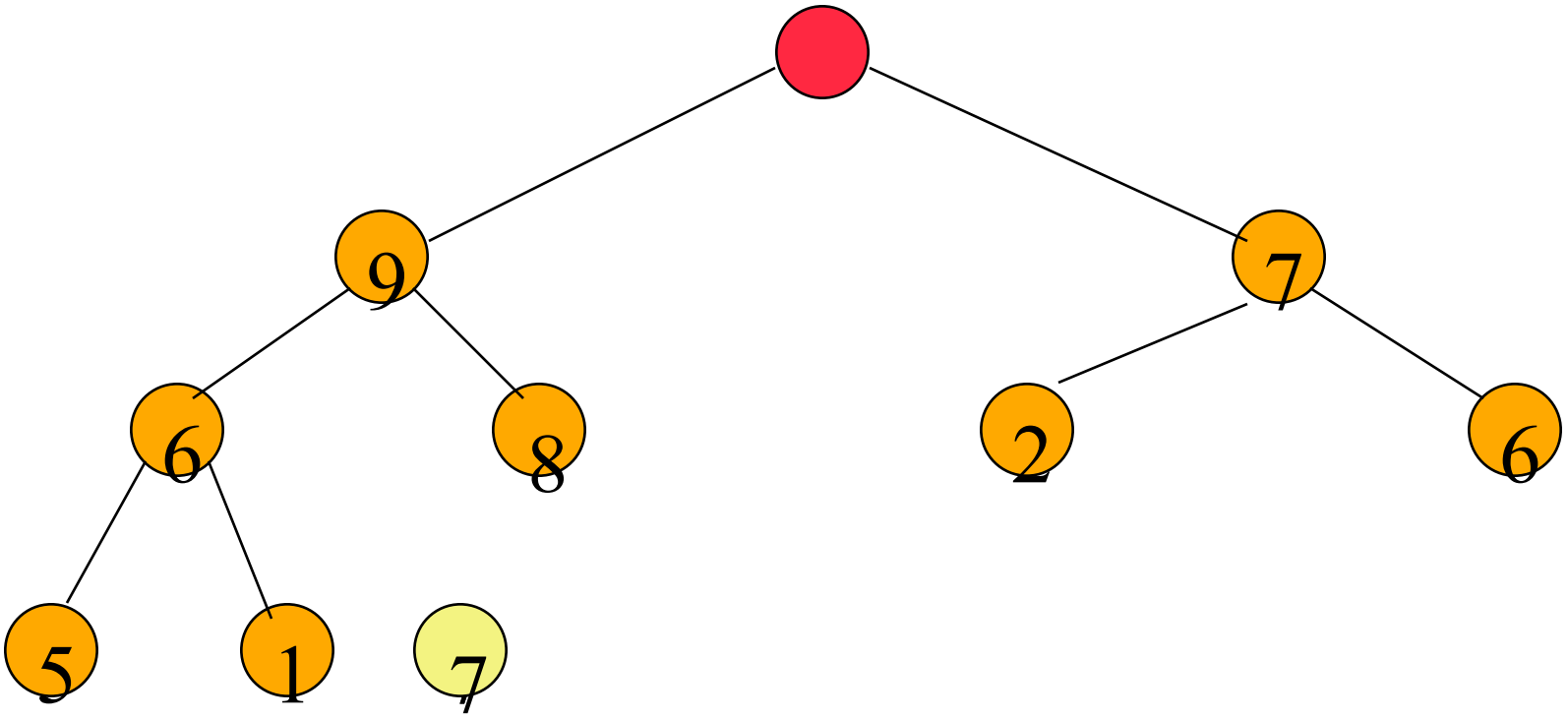


8 .

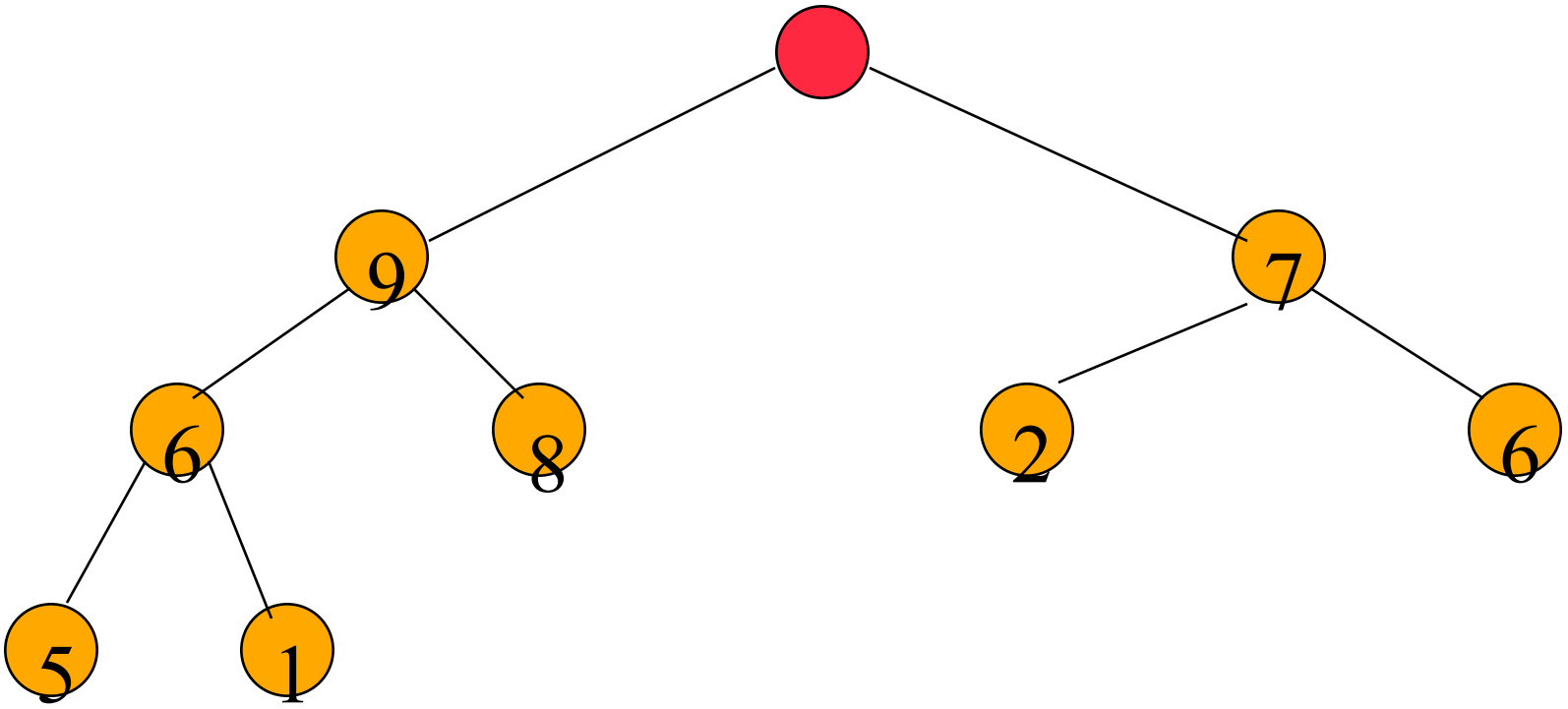


15.

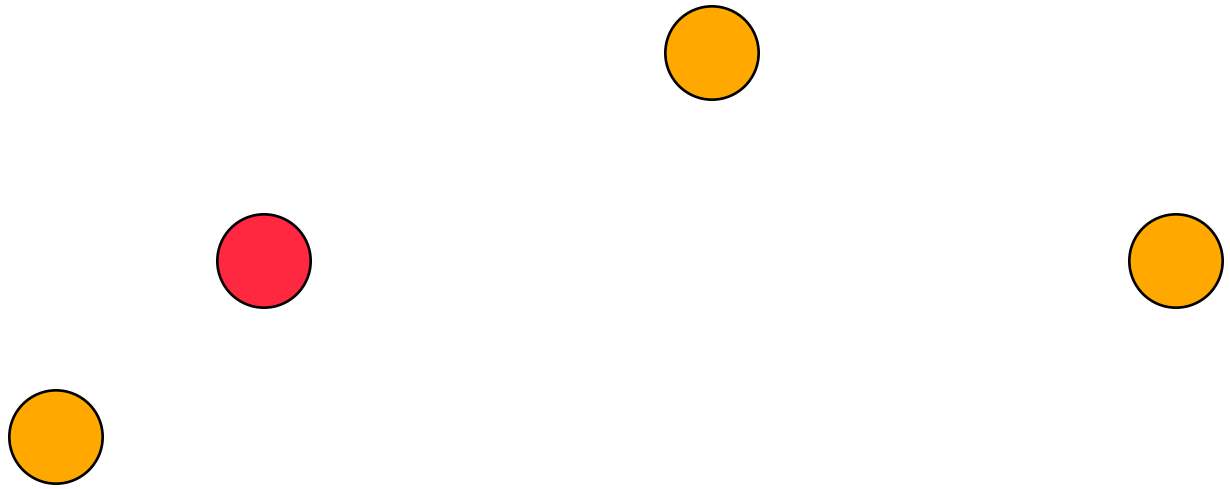

9
7
6
8
2
6
5
1
7



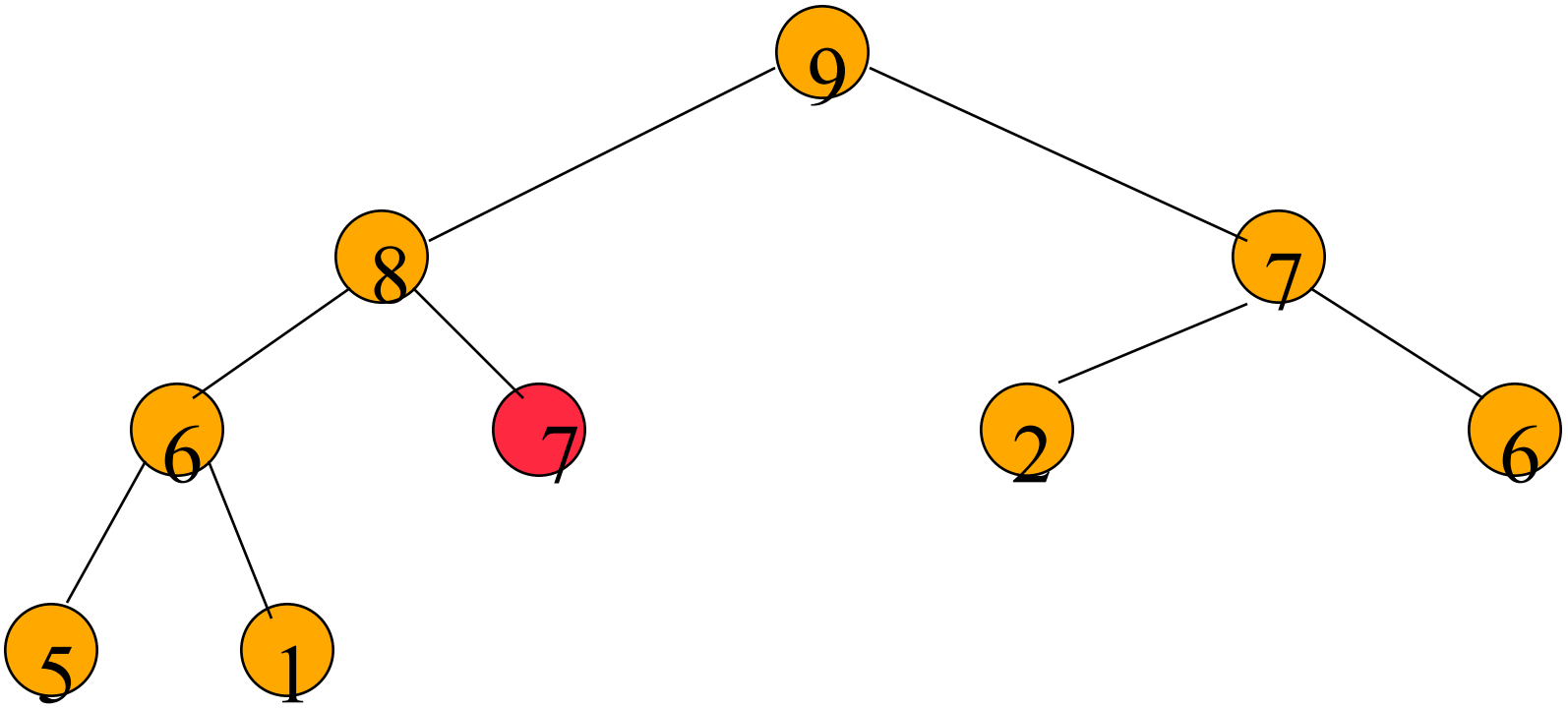
9 .



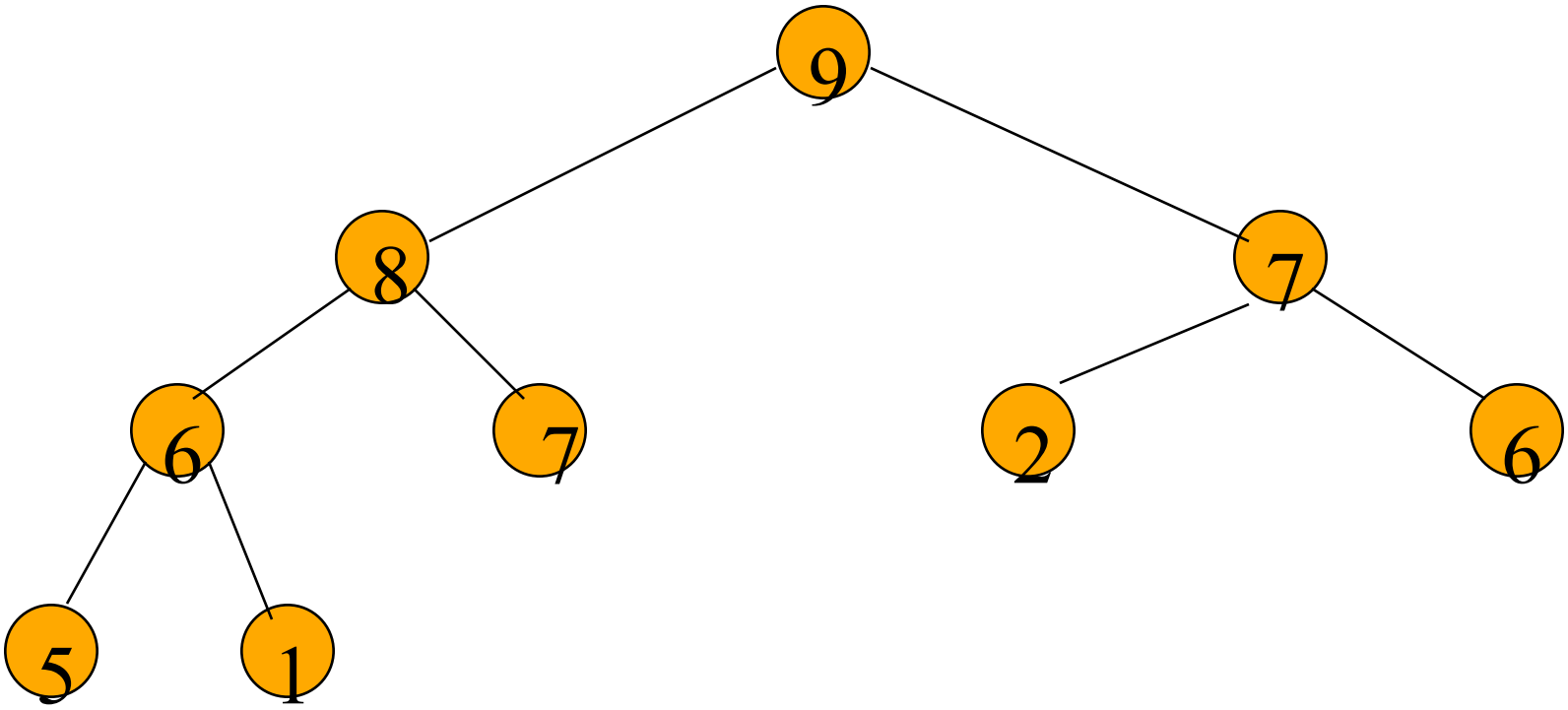
7.



7.



7.


9
8
7
6
7
2
6
5
1

(　　).

< >

< >:: ()

// .

(()) . .

1 . () //

//

= --

//

= 1 //

= 2 //

(<=)

//

(< < +1)

++

//

(>=) //

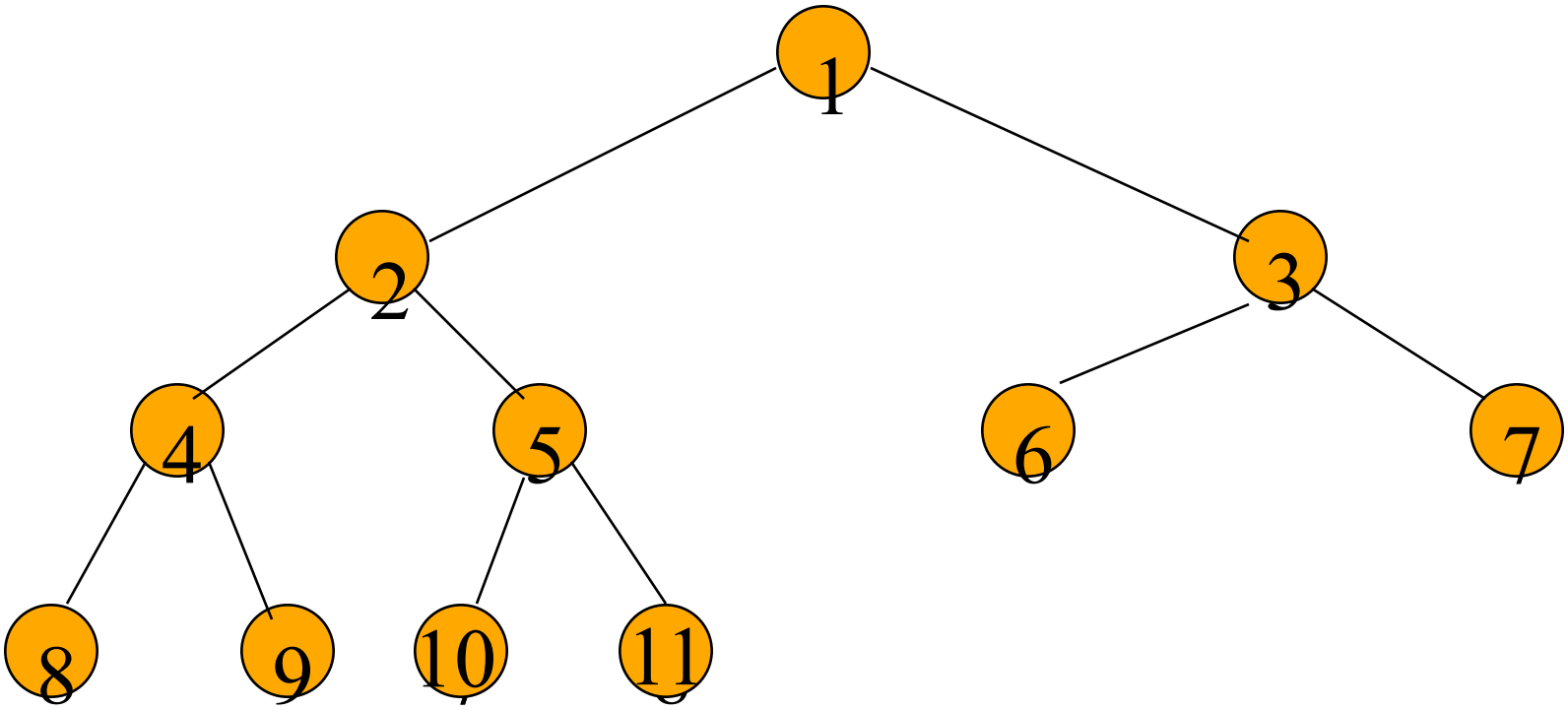
//

= //

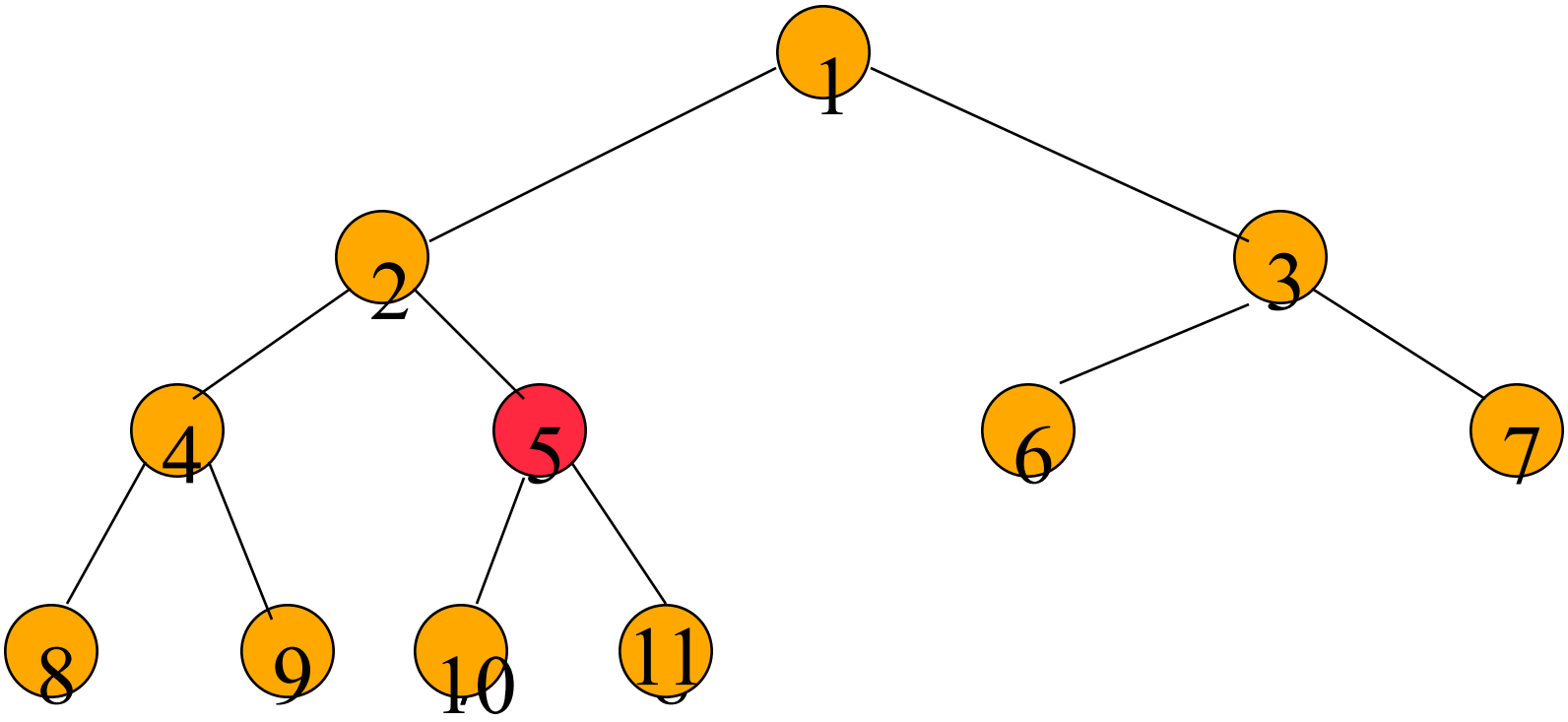
= *=2 //

}

=

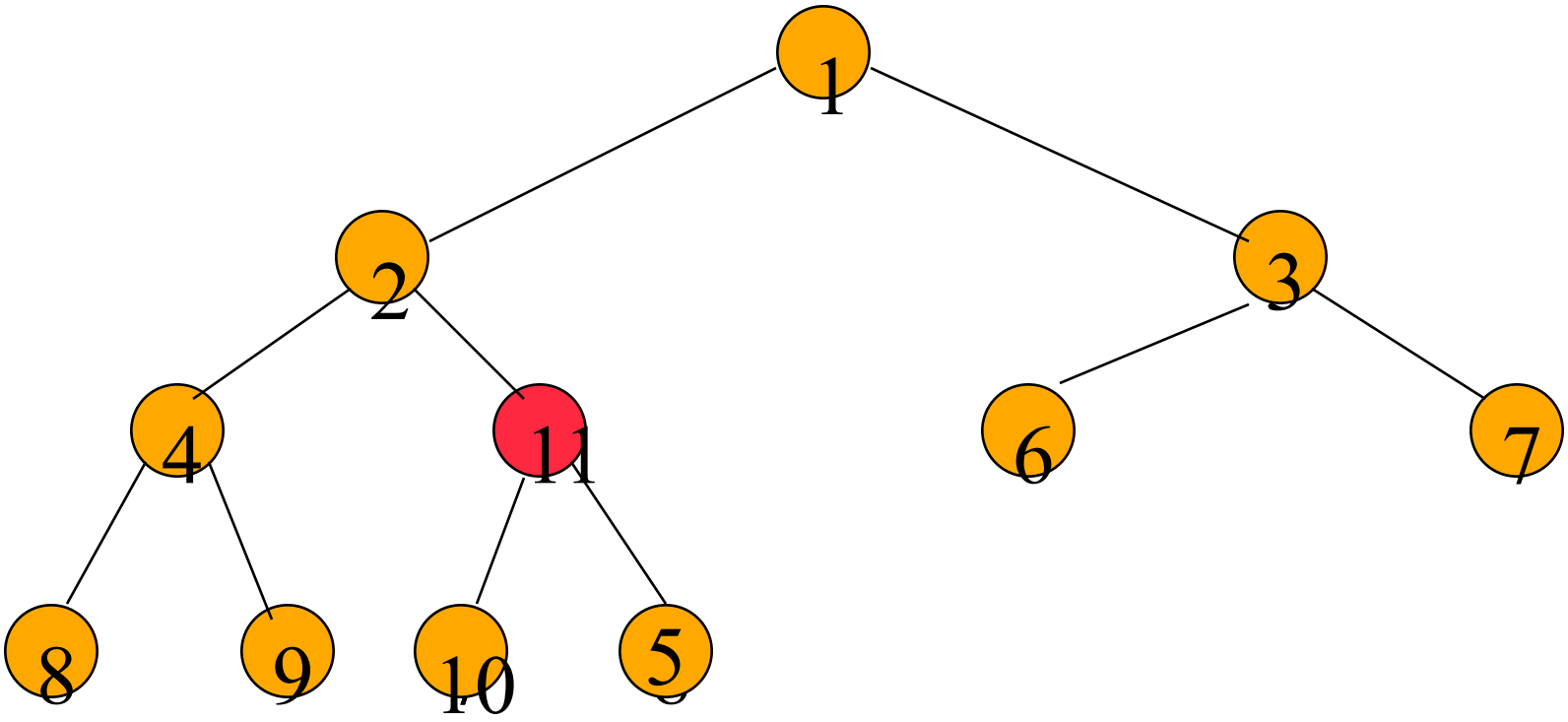


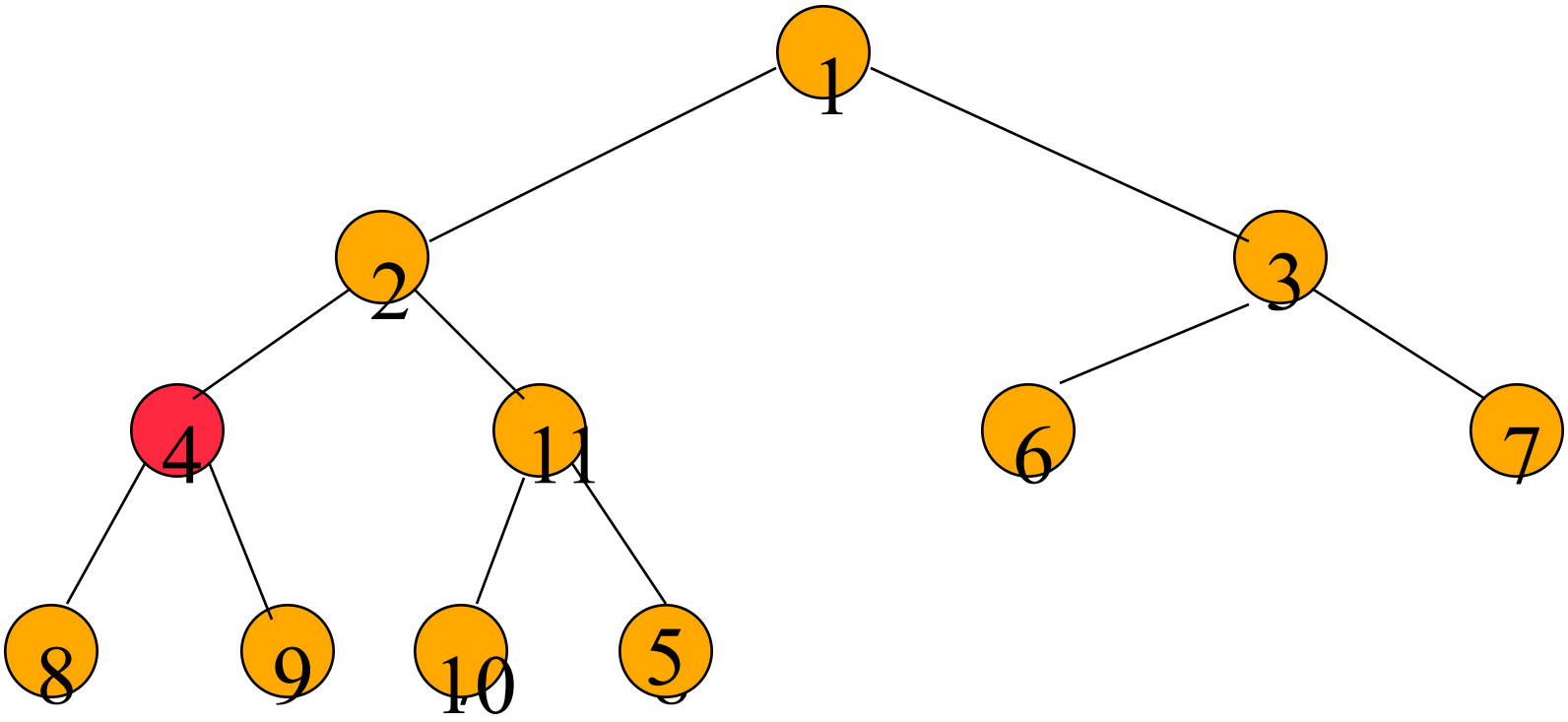
= -, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

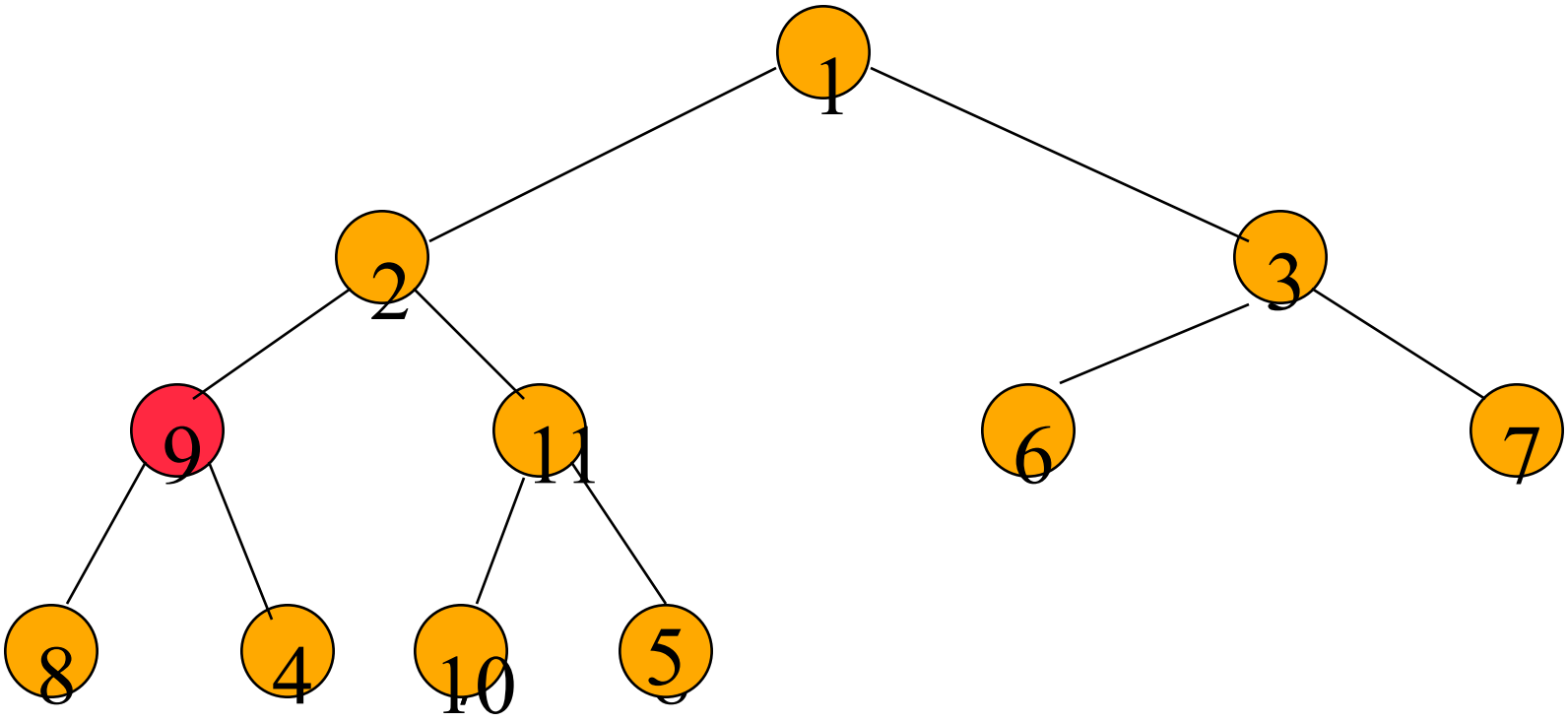


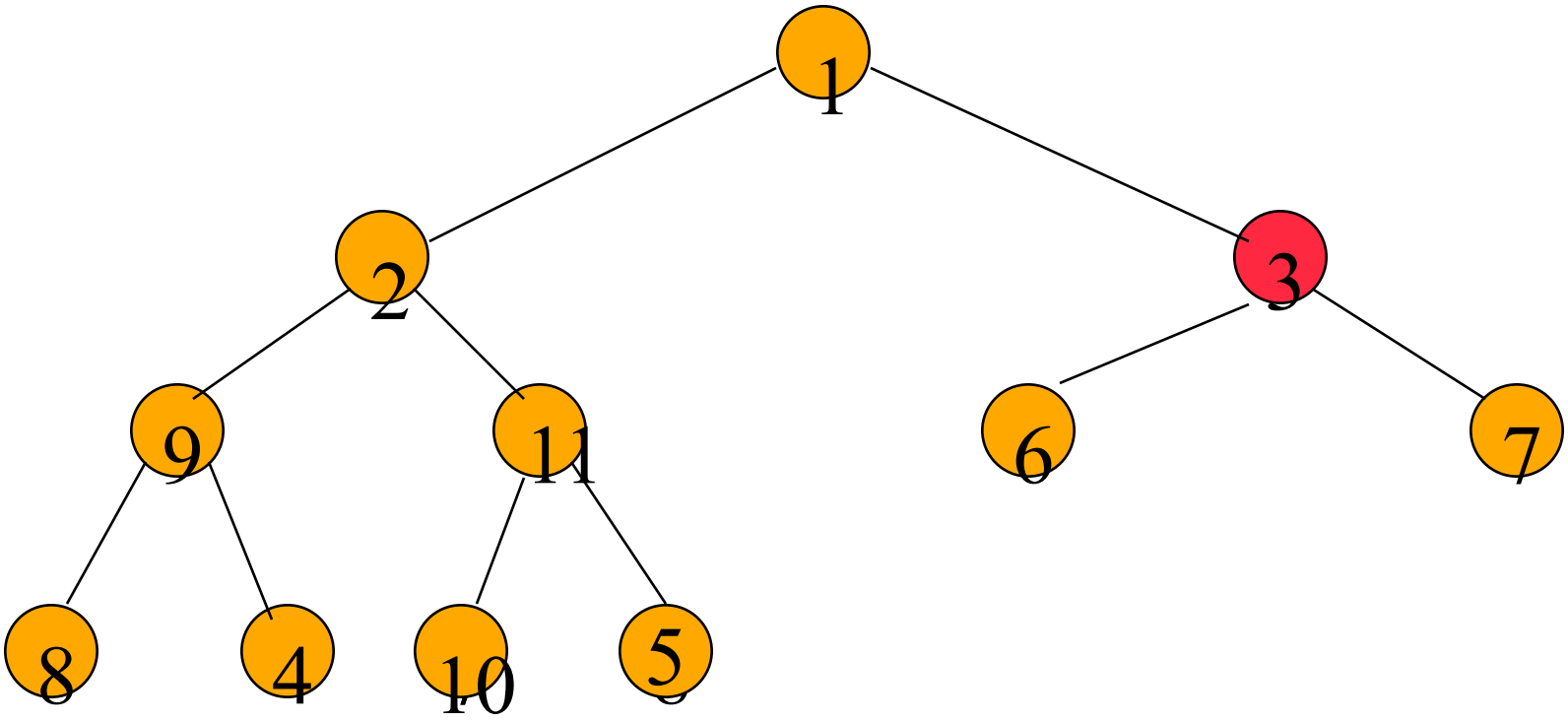
.

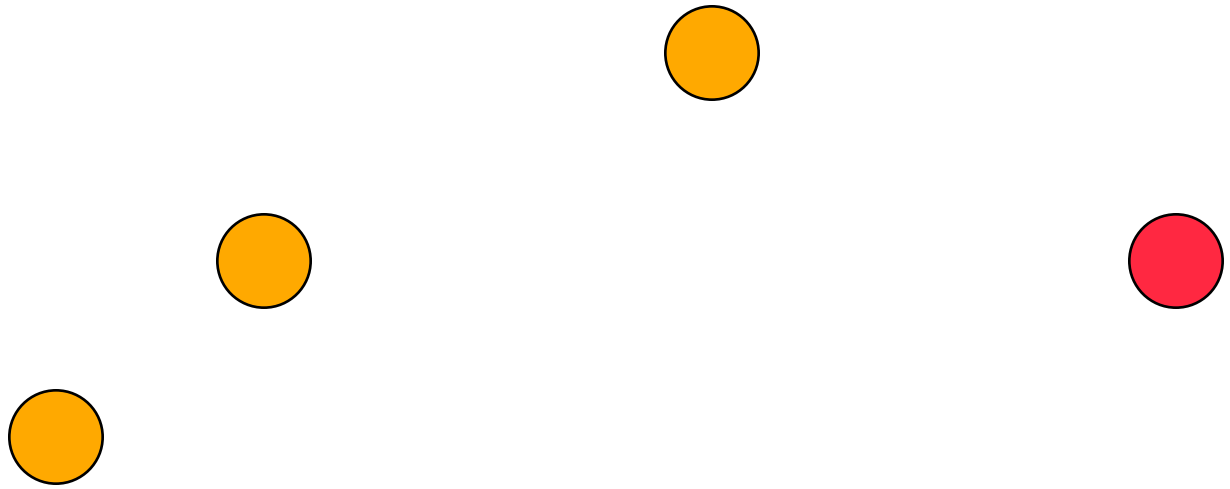
/2.

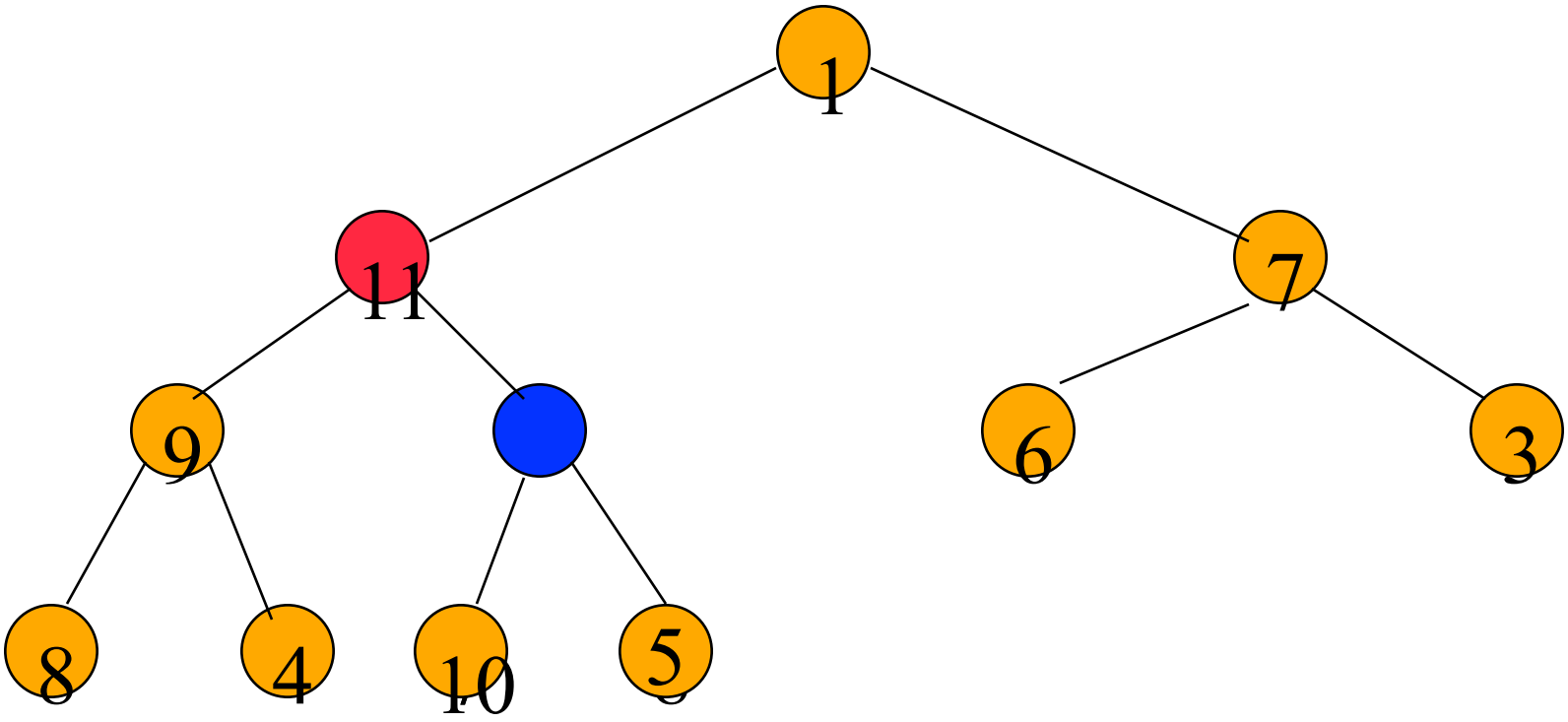

1
2
3
4
11
6
7
8
9
10
5


1
2
3
4
11
6
7
8
9
10
5

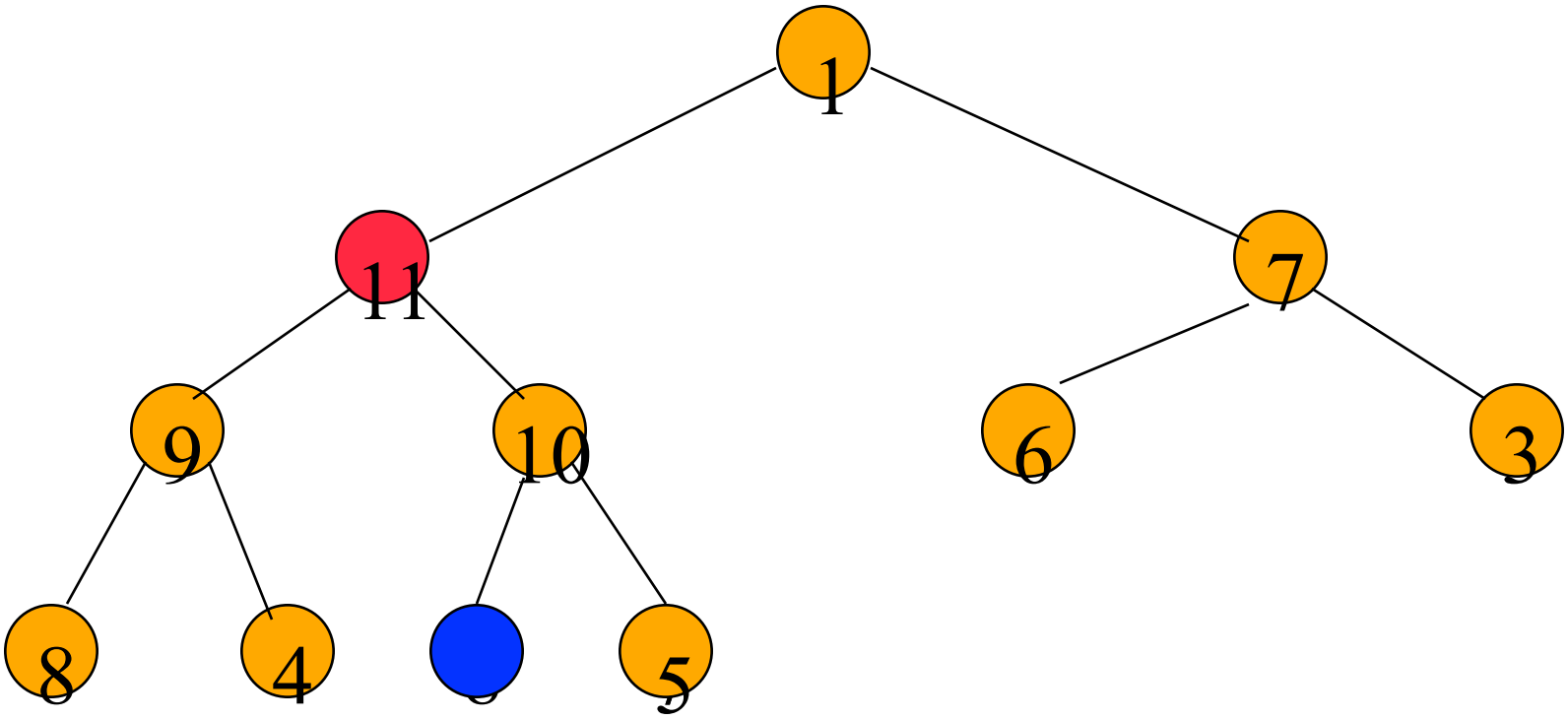

1
2
3
9
11
6
7
8
4
10
5


1
2
3
9
11
6
7
8
4
10
5

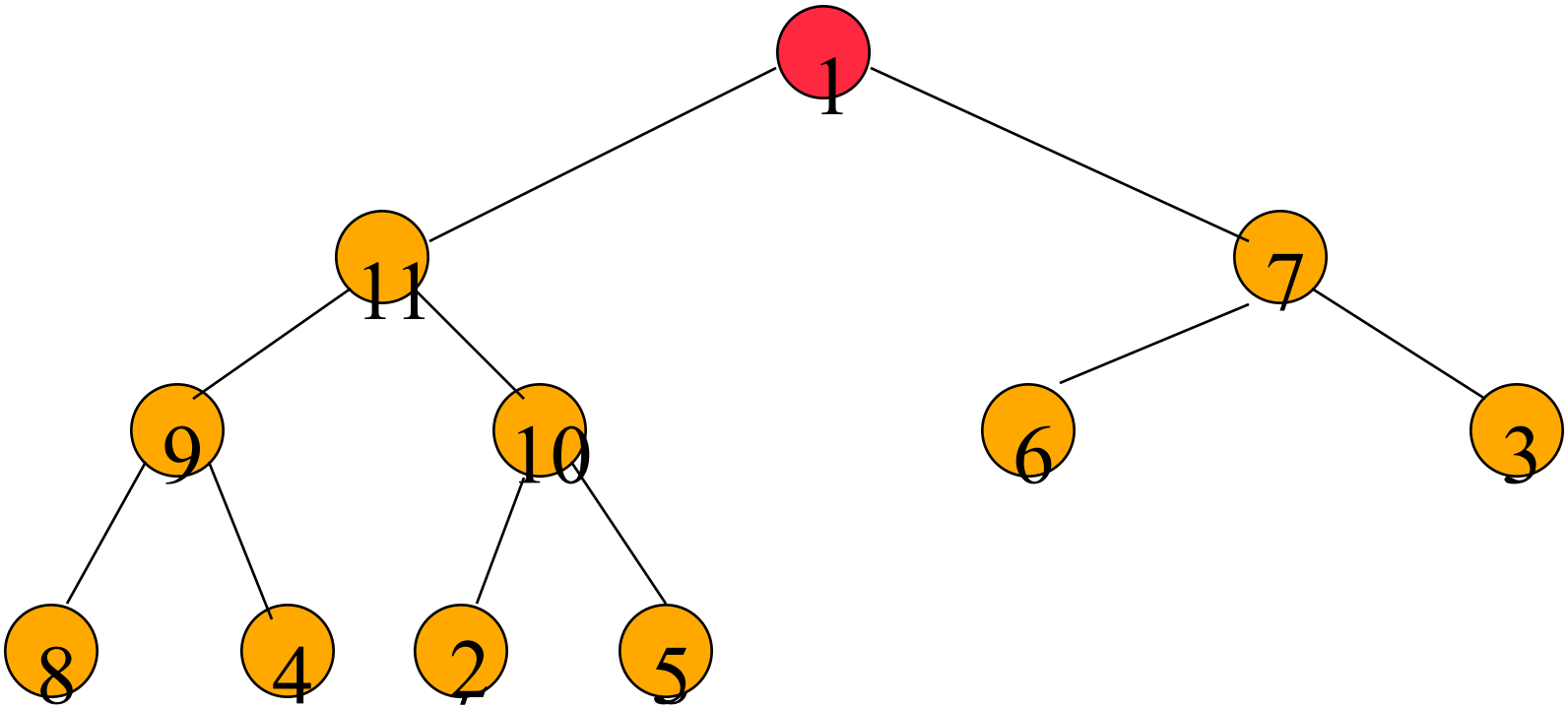




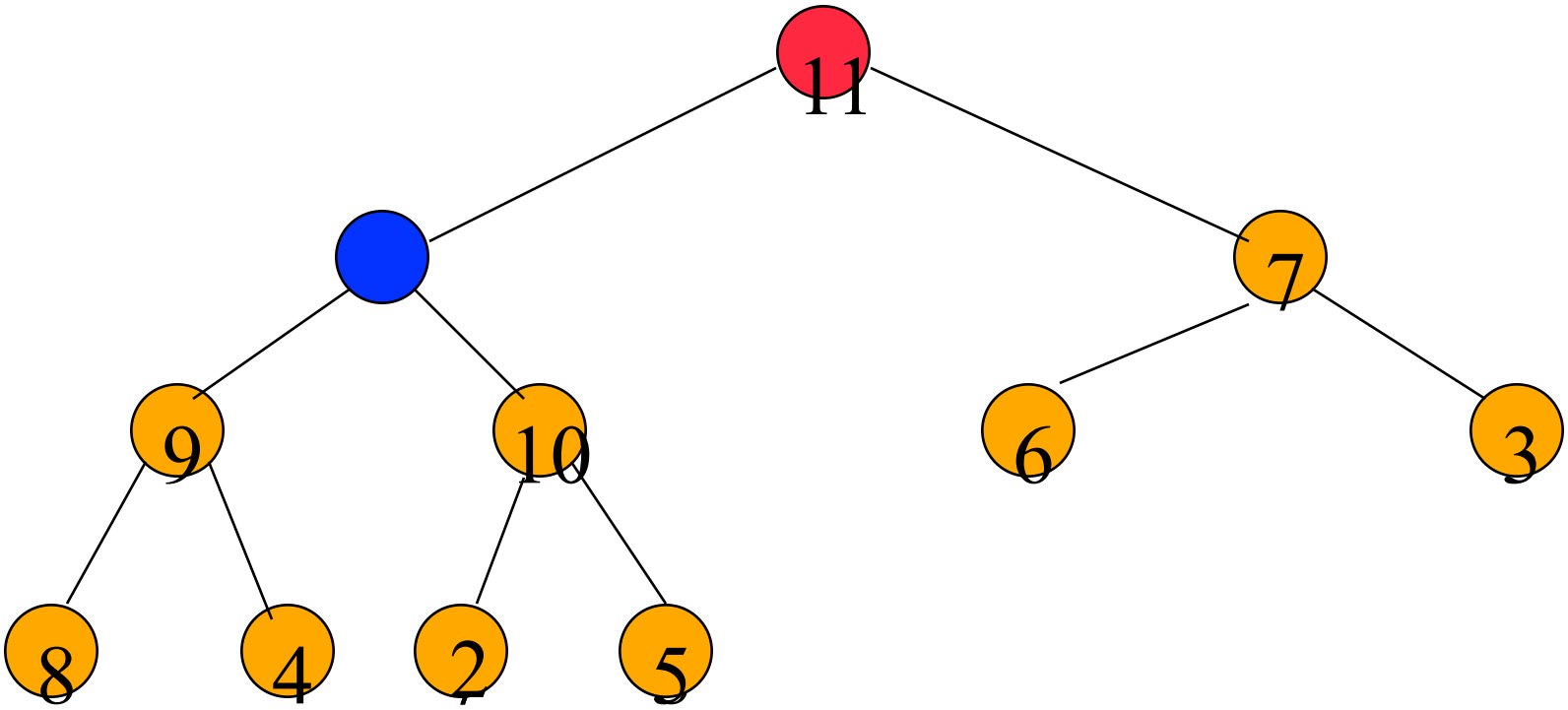
2.



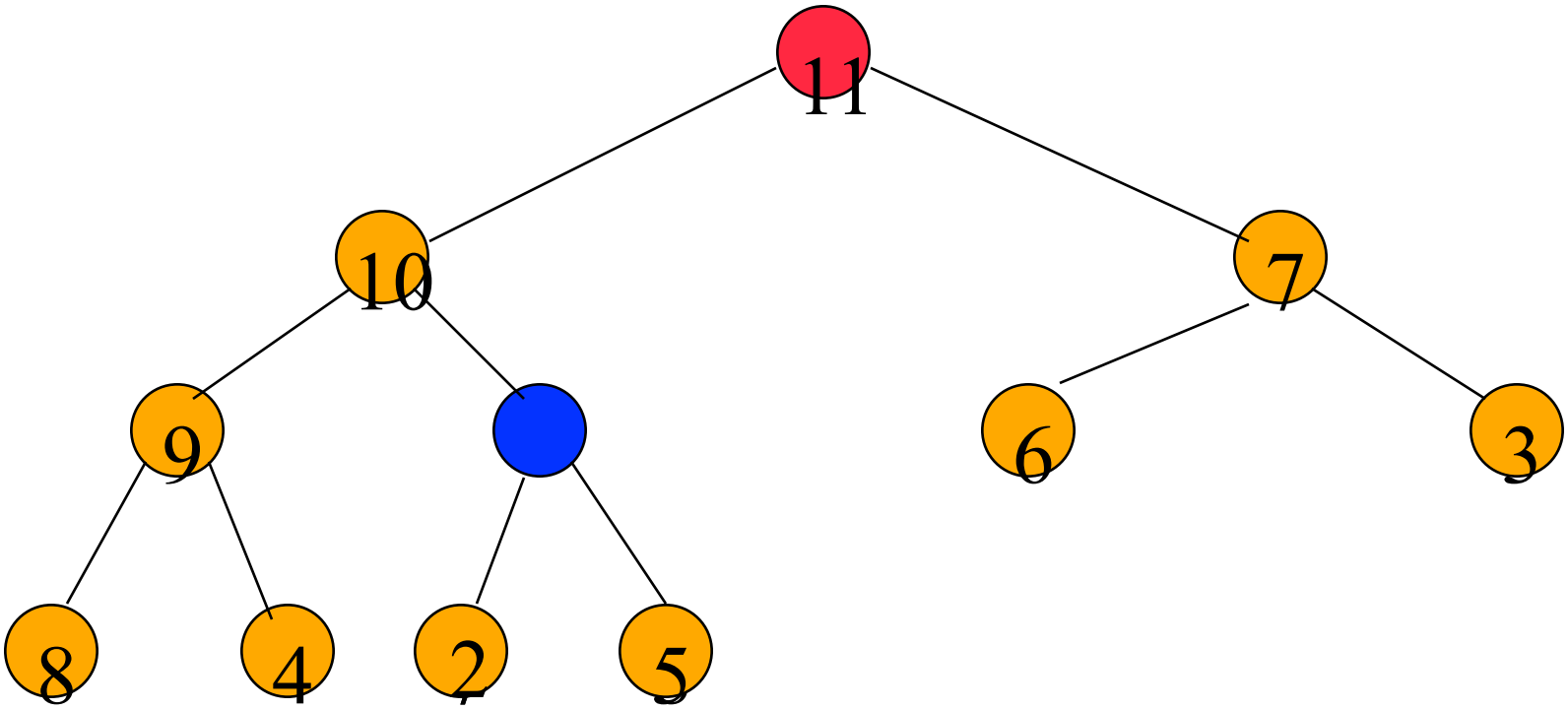
2.



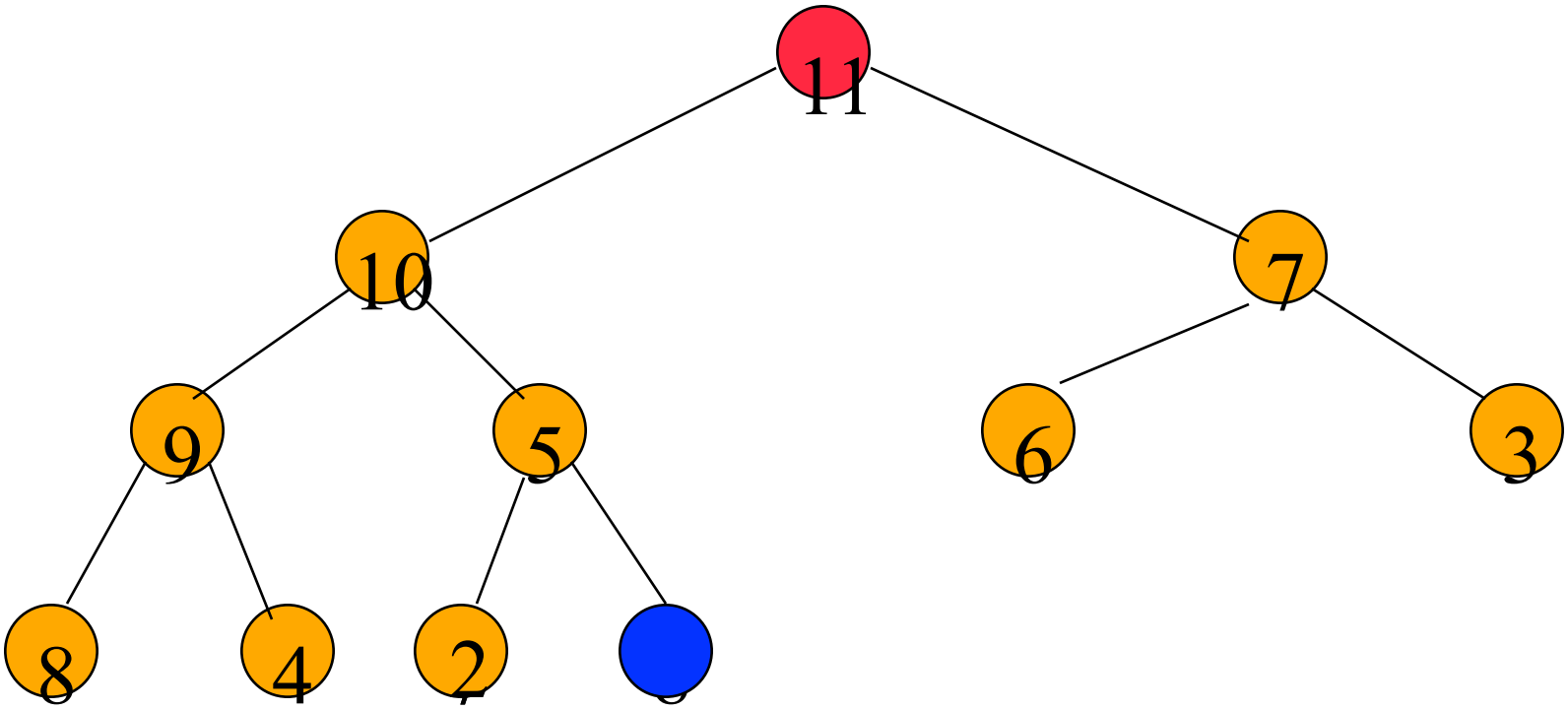
1.



1.



1.



1.

$<= 2^{-1}(- +1) = ( )$.

(1) + (2) + + (-1) = ( ).

:

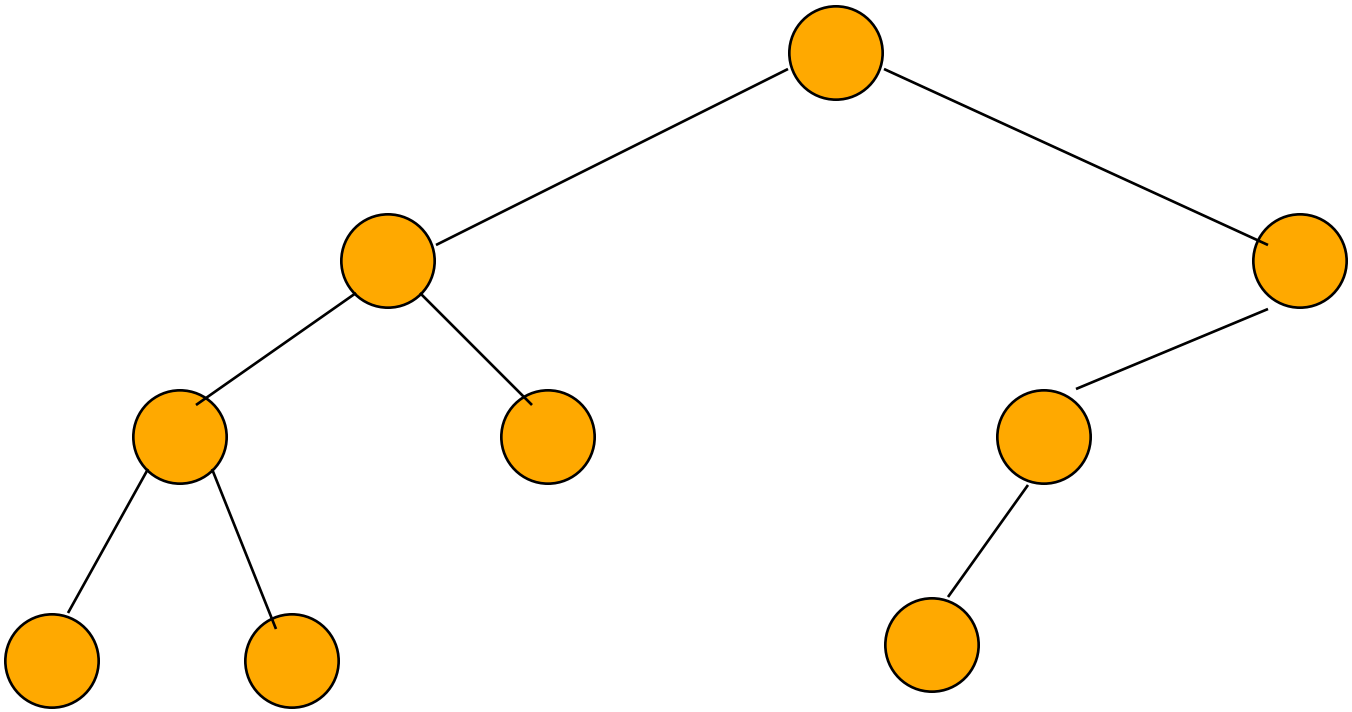
++.

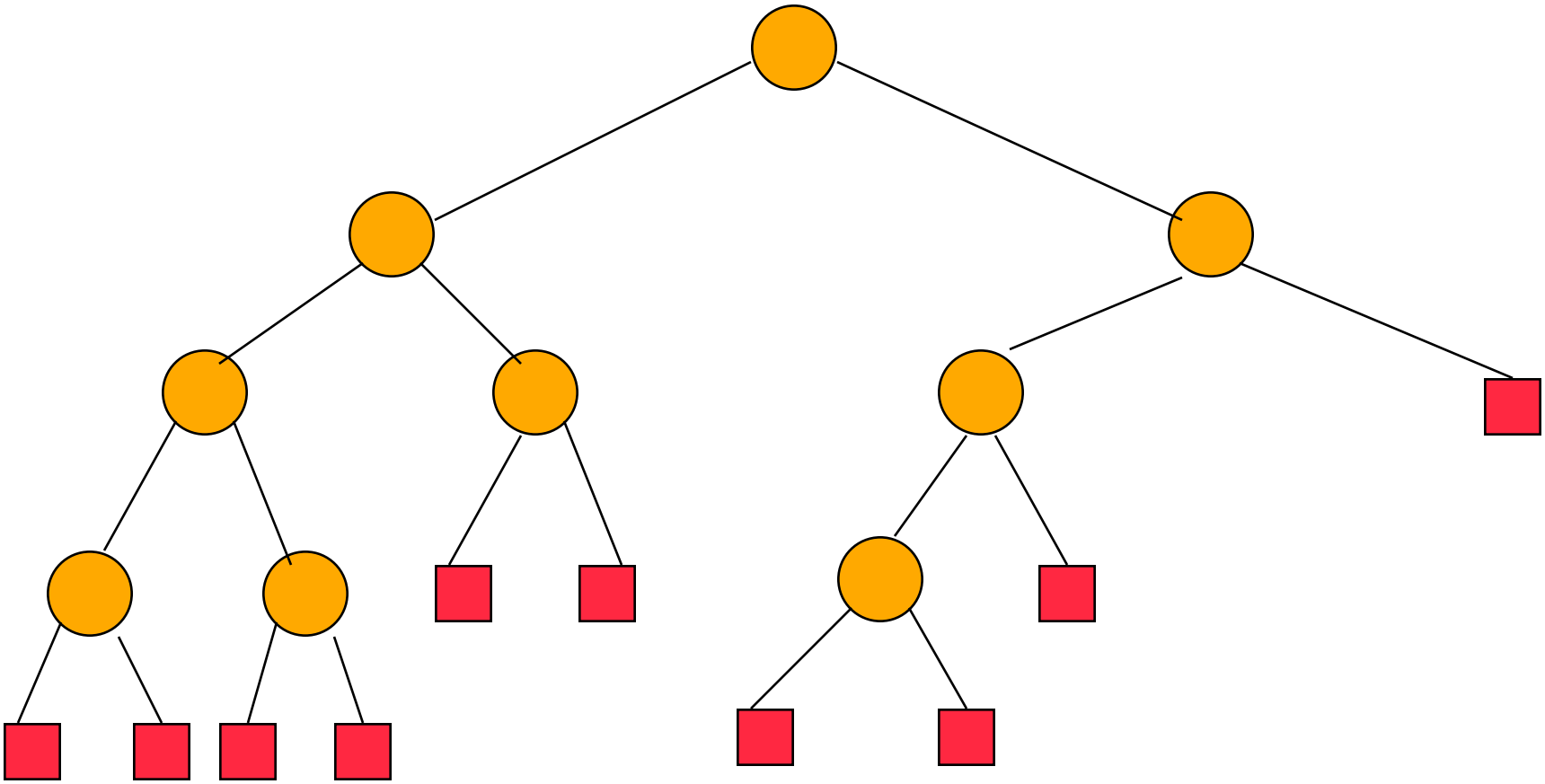
.

.

-
- ( )
-

( ) .





+1

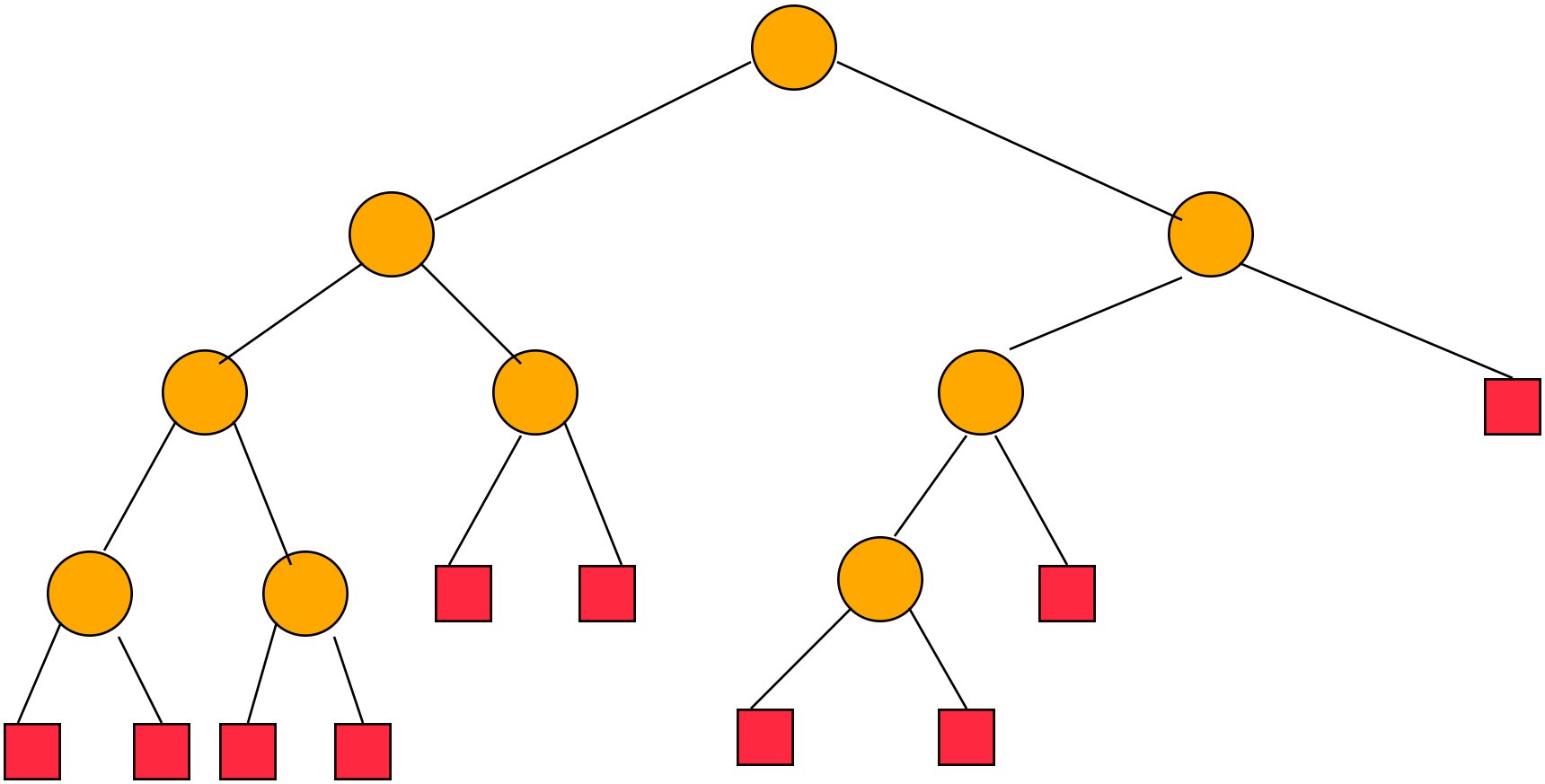
()

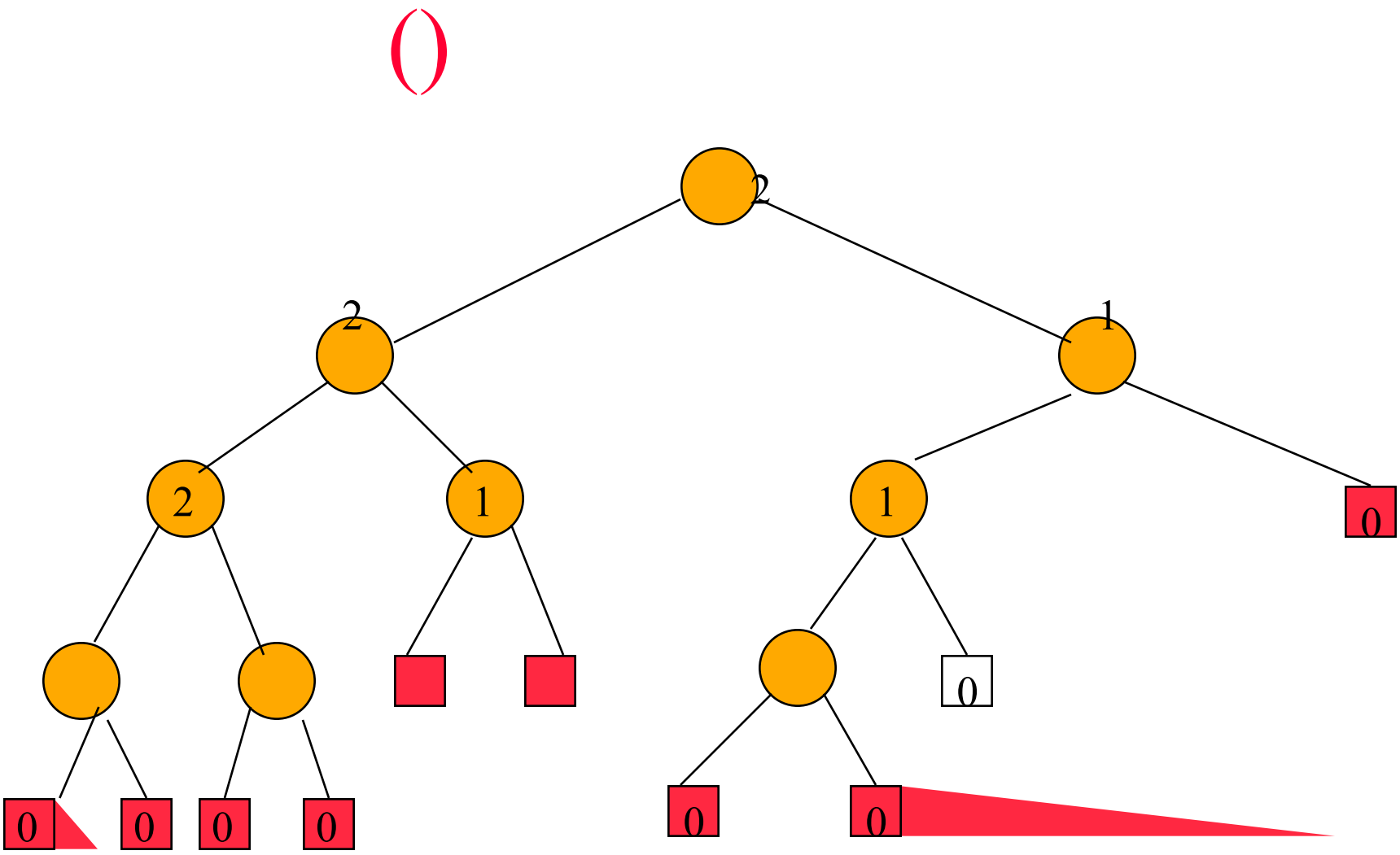
,

( )

.

()




()
2
2
1
2
1
1
0
0
0
0
0
0
0
0

()

, ( ) = 0.

,

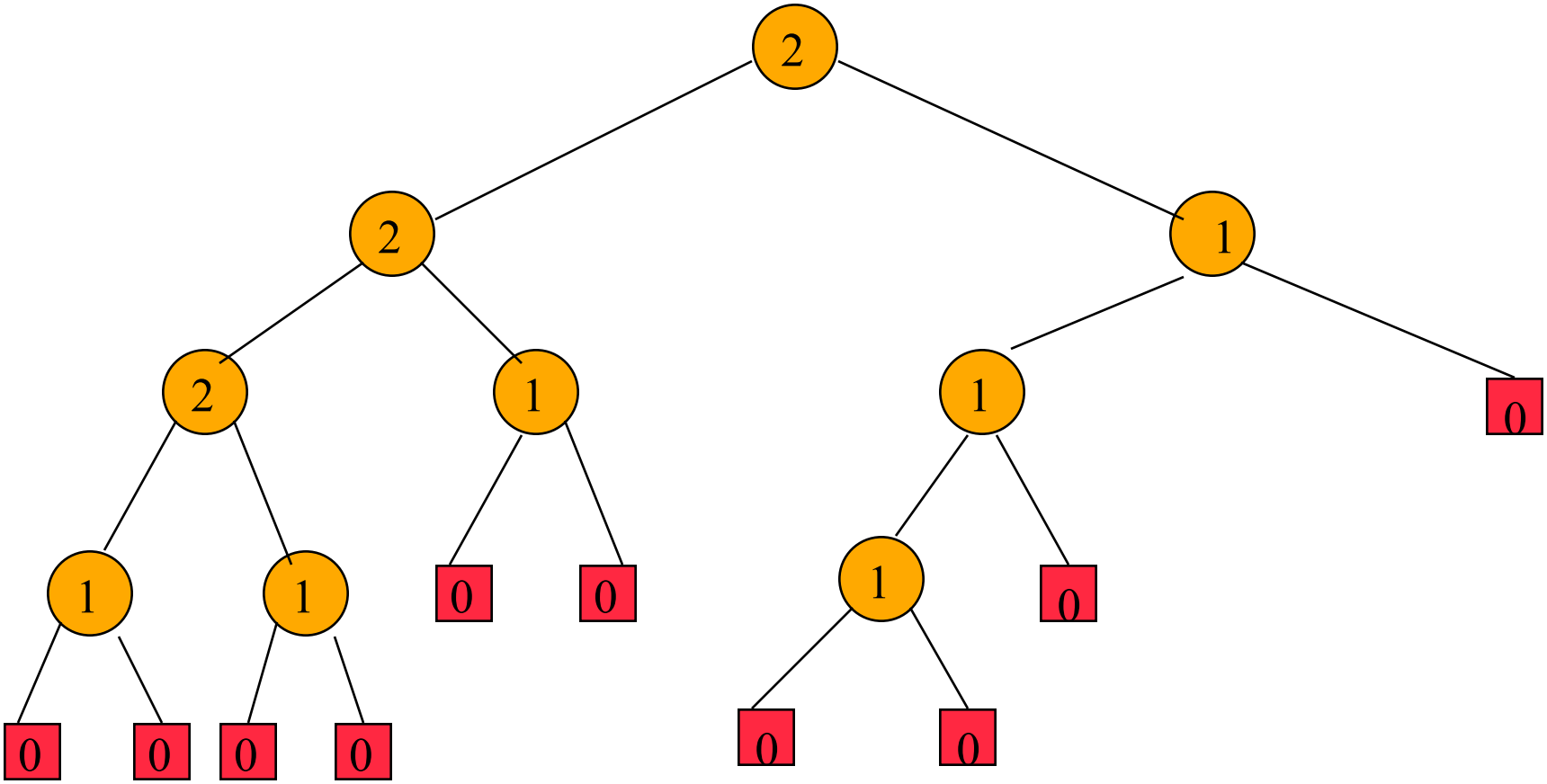
( ) = (( )),

(( )) + 1

( )

,

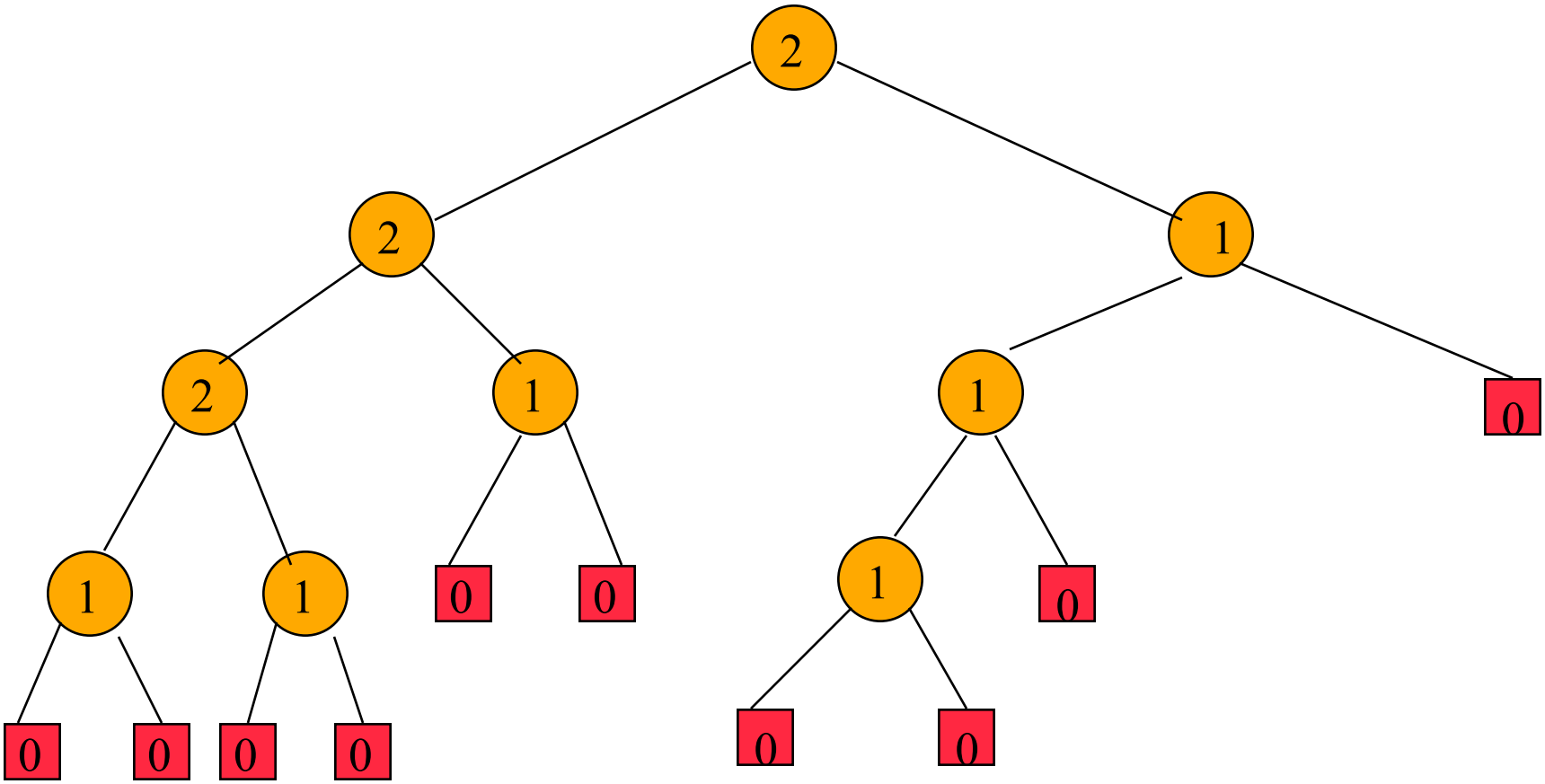
(( )) >= (( ))


2
2
1
2
1
1
0
1
1
0
0
1
0
0
0
0
0
0

1

,

( ).



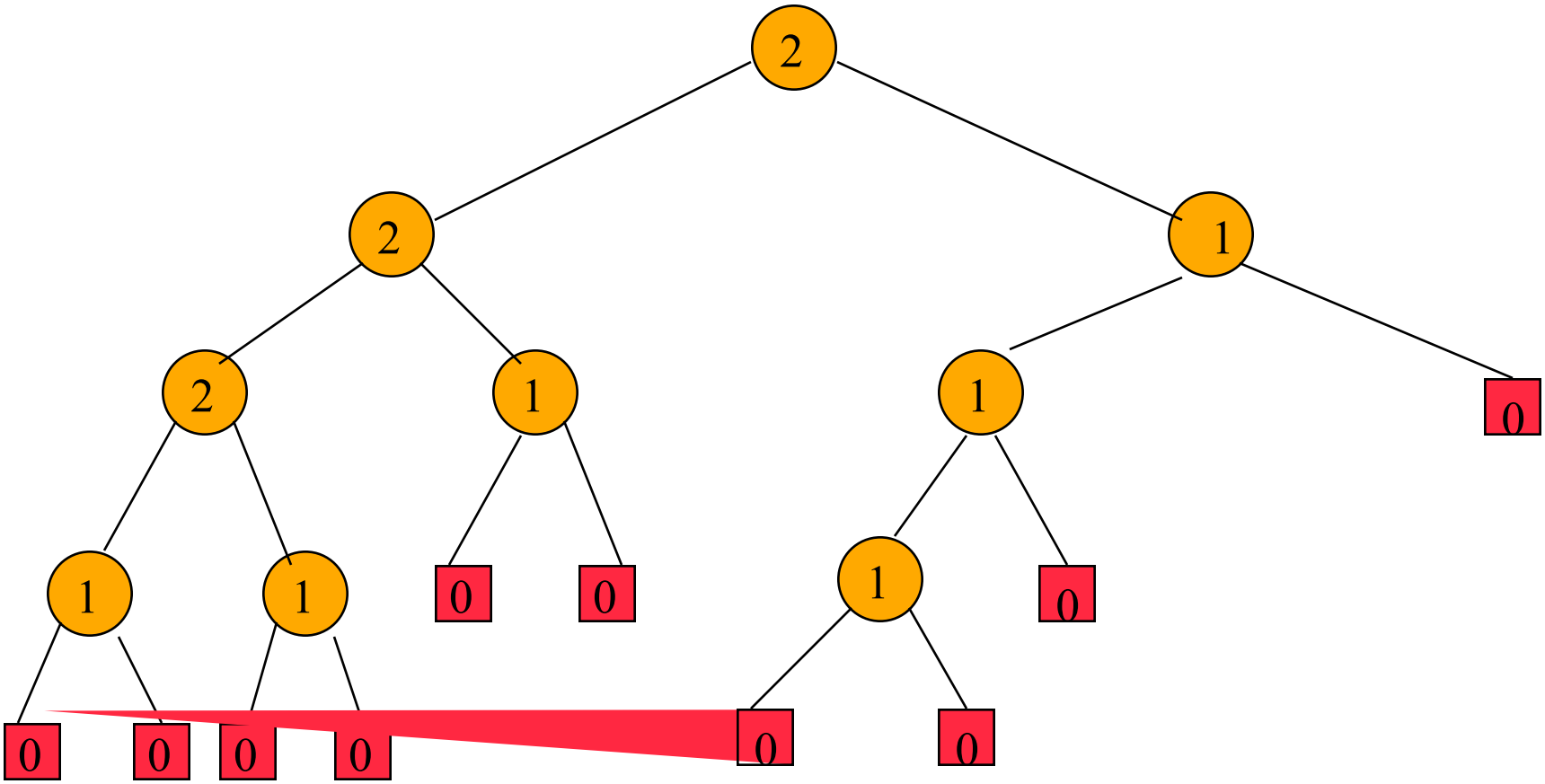
2.

2

$2^{(\quad)} - 1$

1 ( )

.



1 2 .

3

( ),

( )

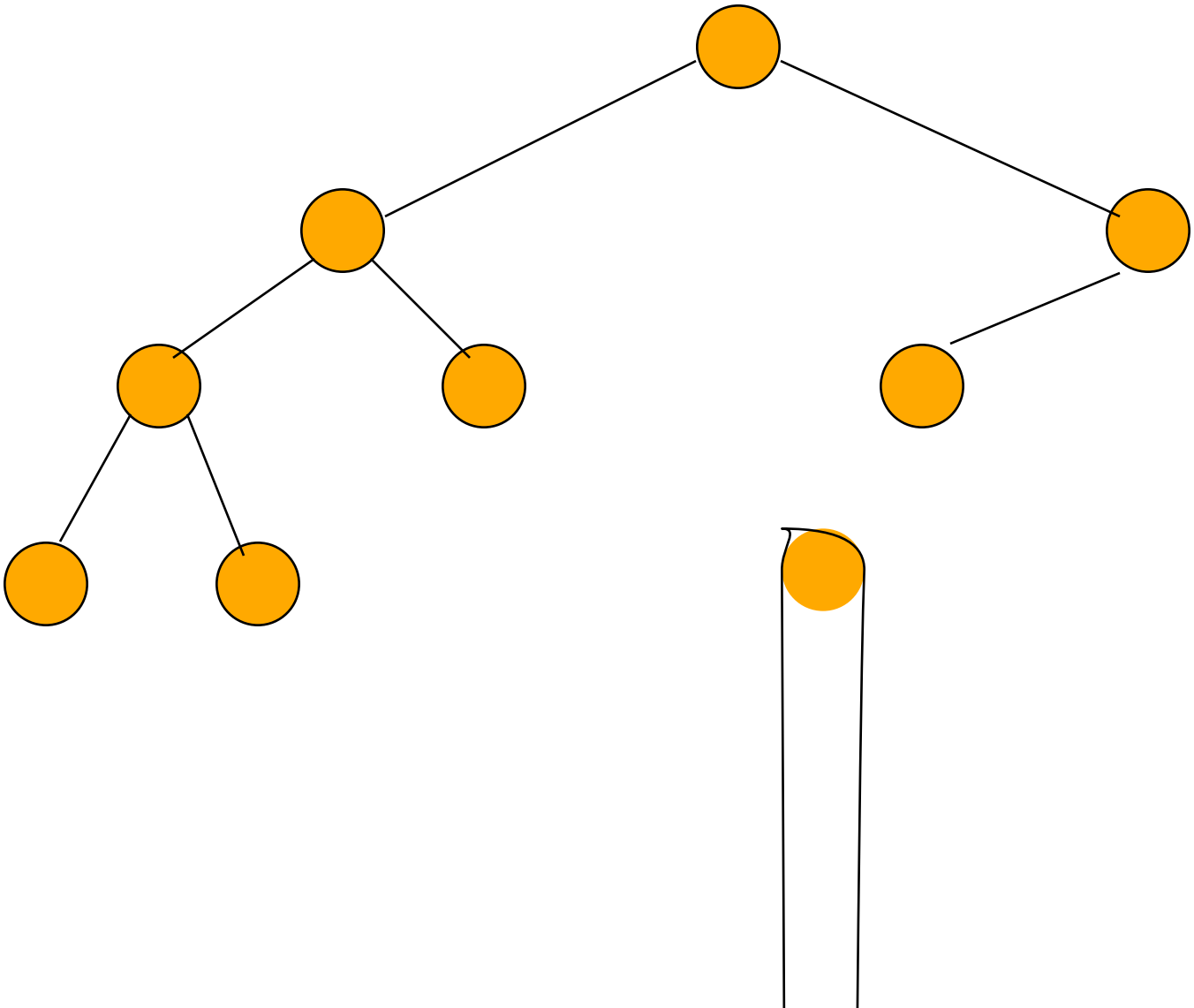
.

2 =>

- >= 2 ( ) 1 => ( ) <= $_2$(+1)

1 =>

( ).

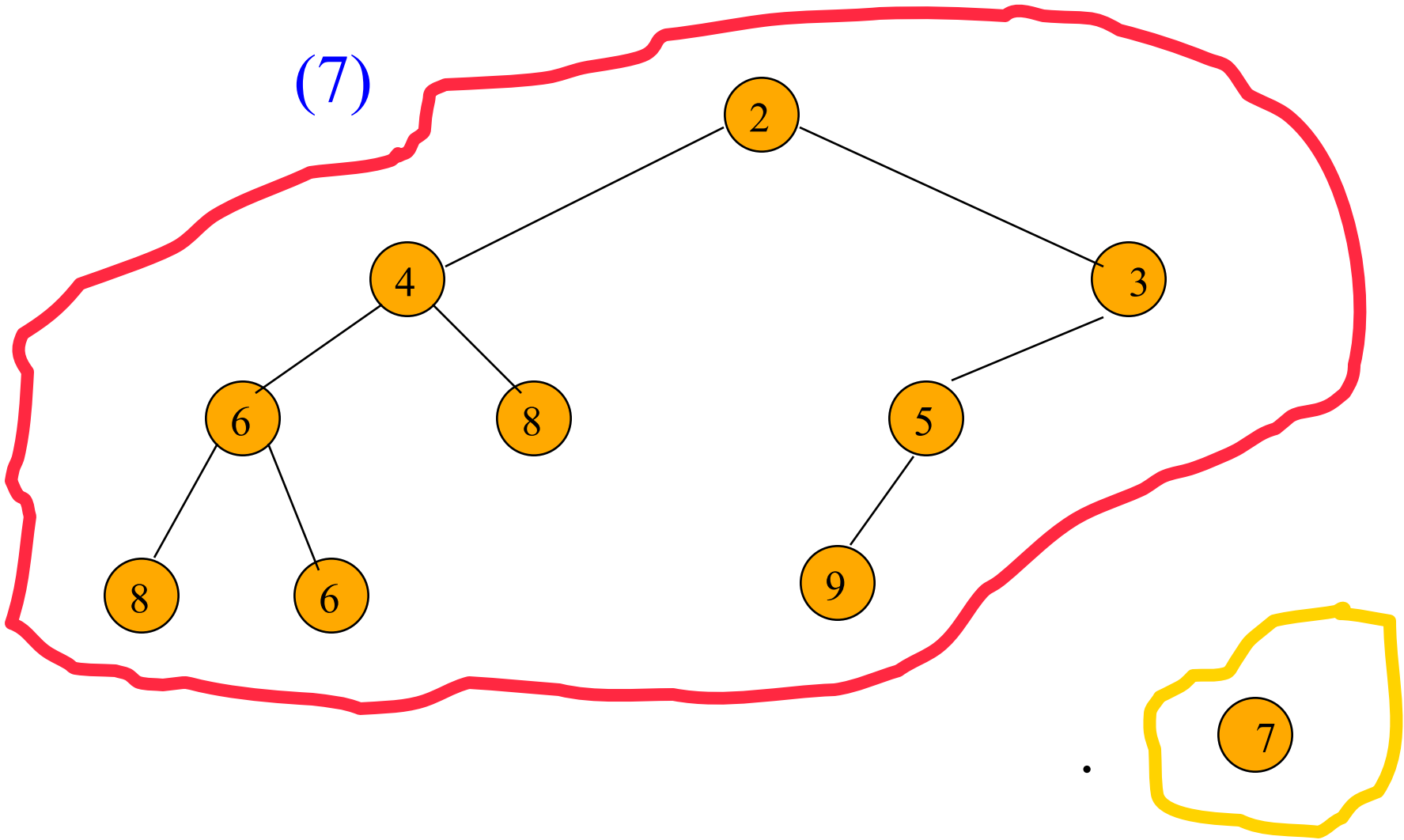


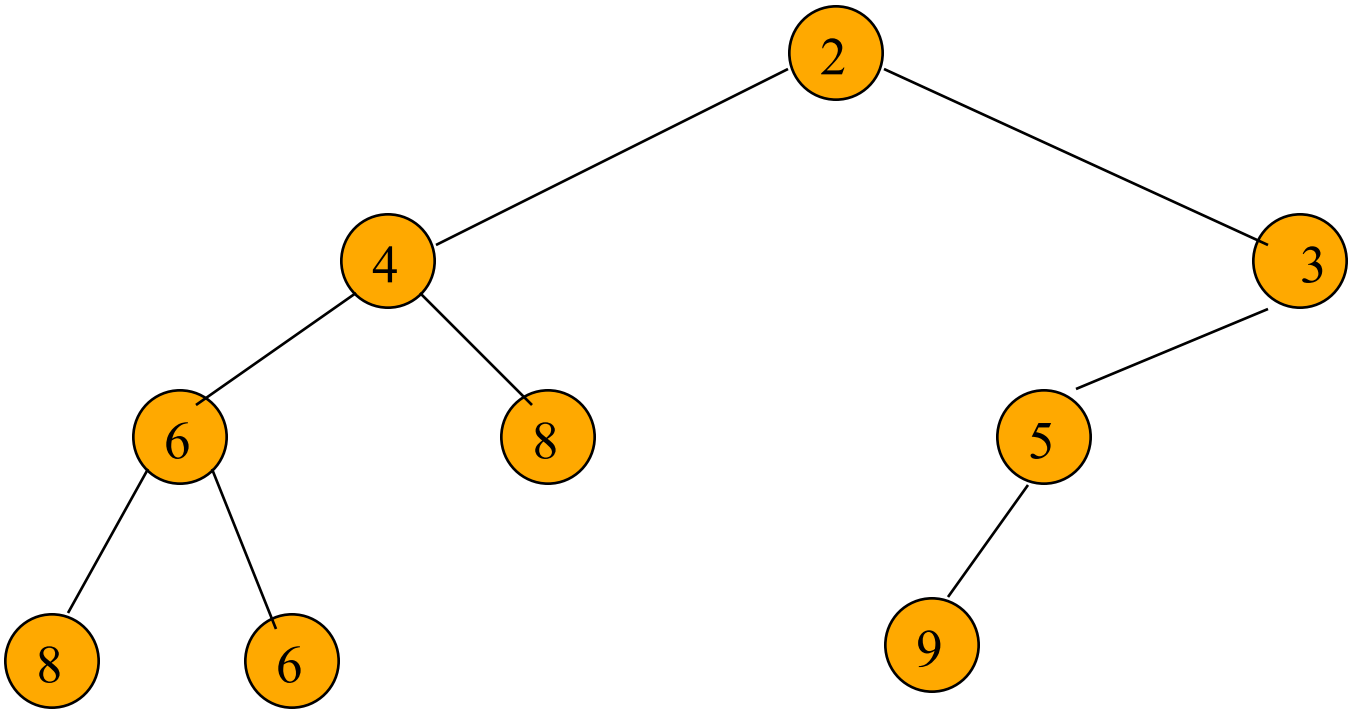
()

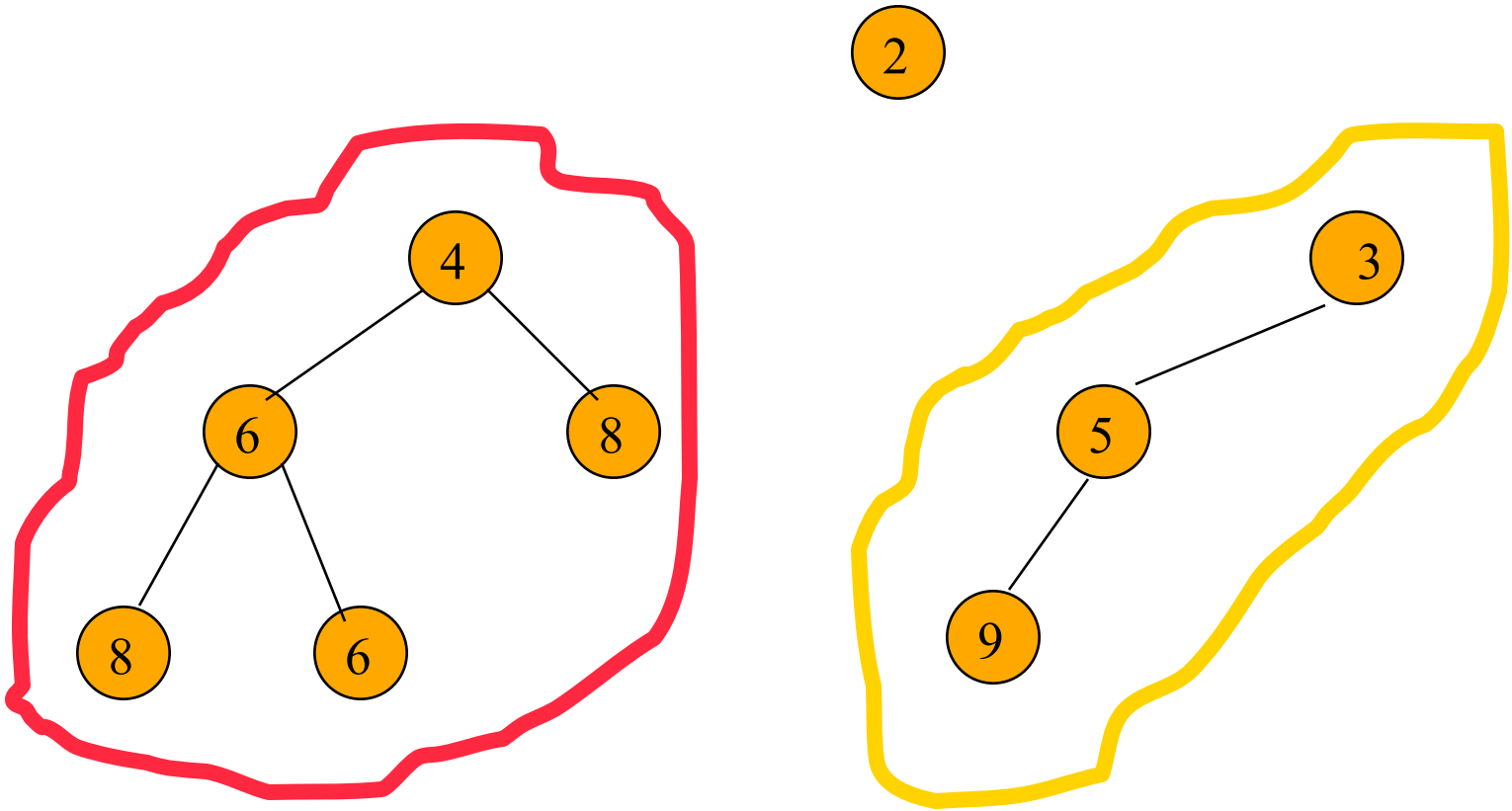
()

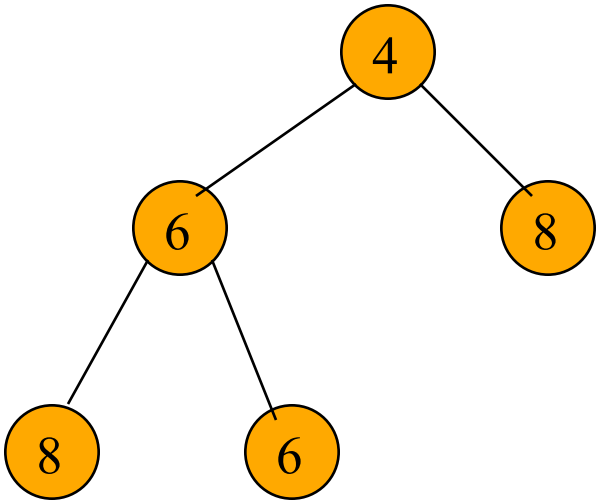
()

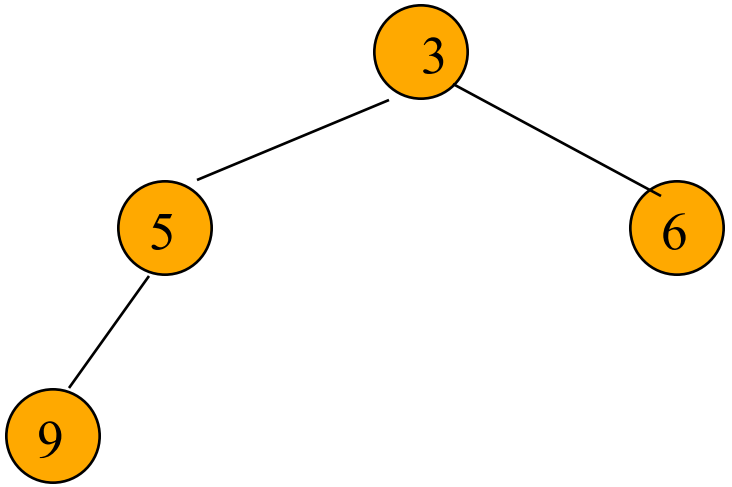
() () ().

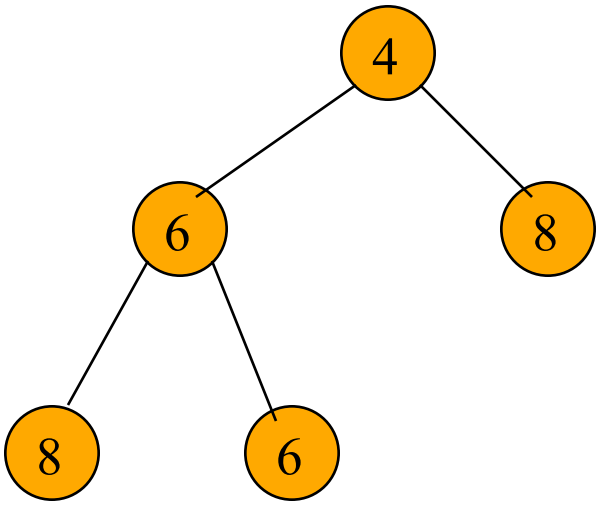

(7)
2
4
3
6
8
5
8
6
9
7

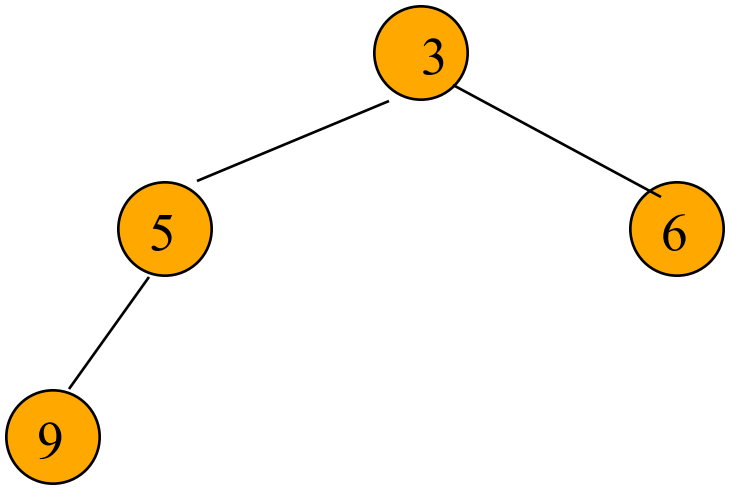

2
4
3
6
8
5
8
6
9

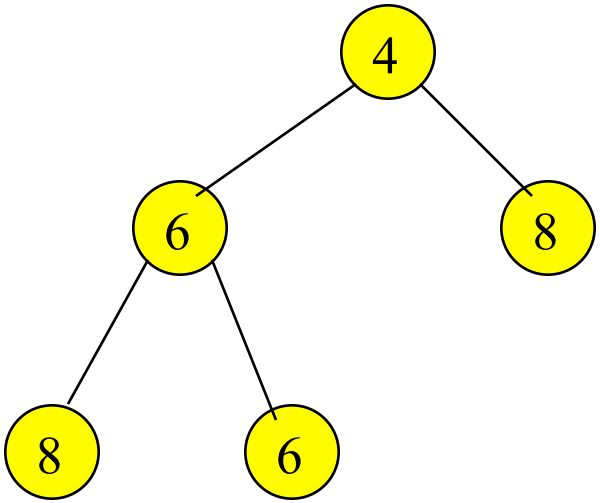

2
4
6
8
8
6
3
5
9

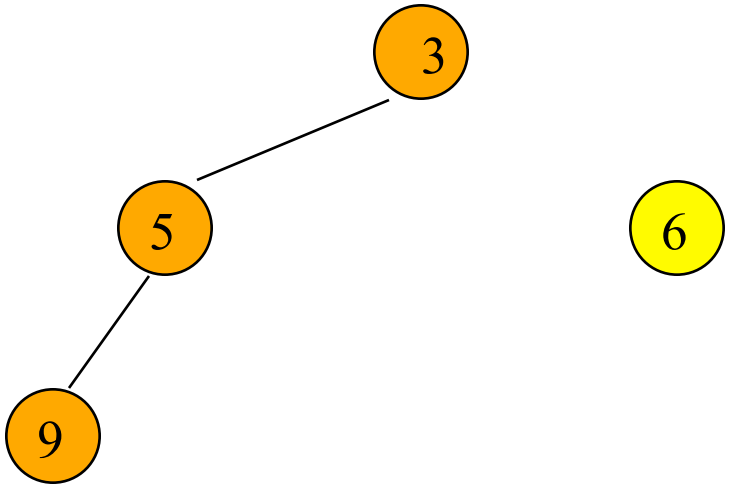

4
6
8
8
6

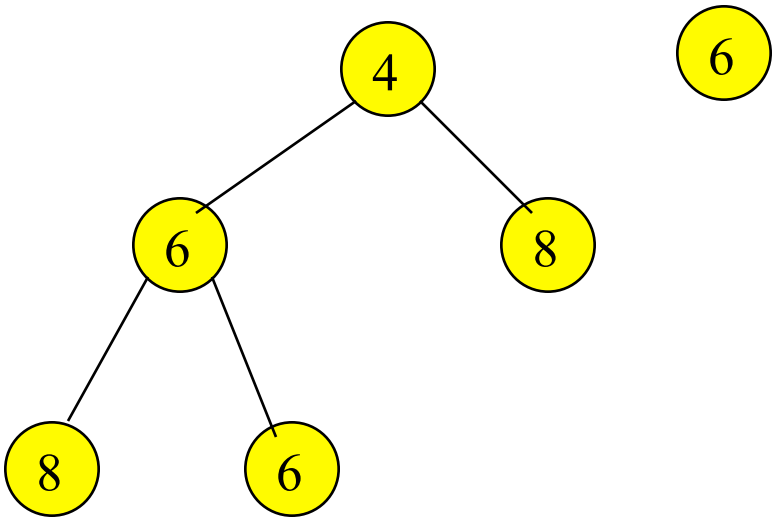

3
5
6
9


4
6
8
8
6


3
5
6
9


4
6
8
8
6


3
5
6
9


4
6
8
8
6
6

8

6

.

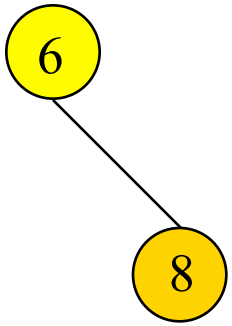
6

. ,

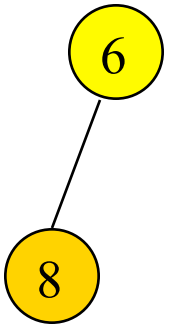
.

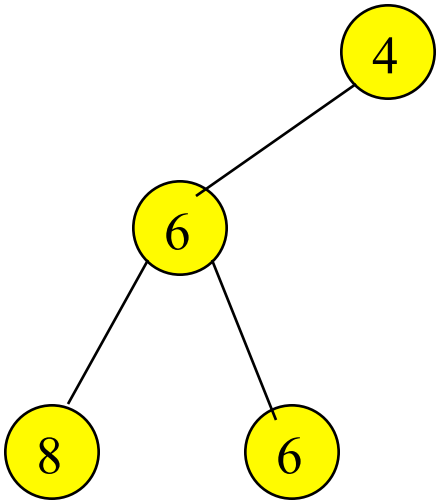

8
6

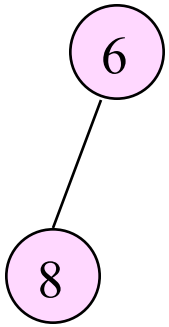
.

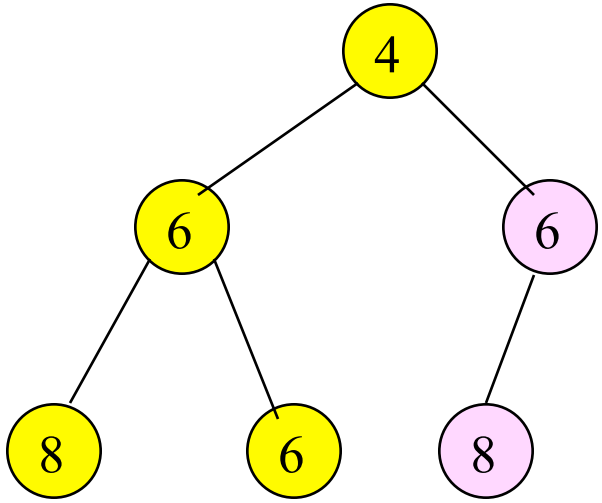

6
8

( ) < ( ).


6
8

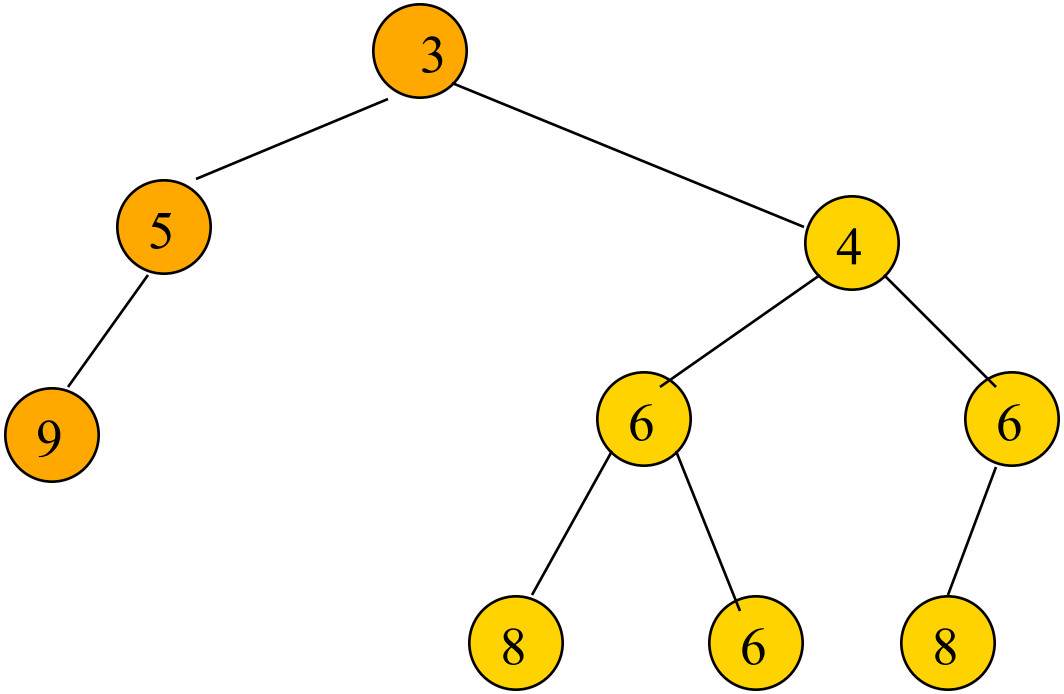

4
6
8
6


6
8


4
6
6
8
6
8

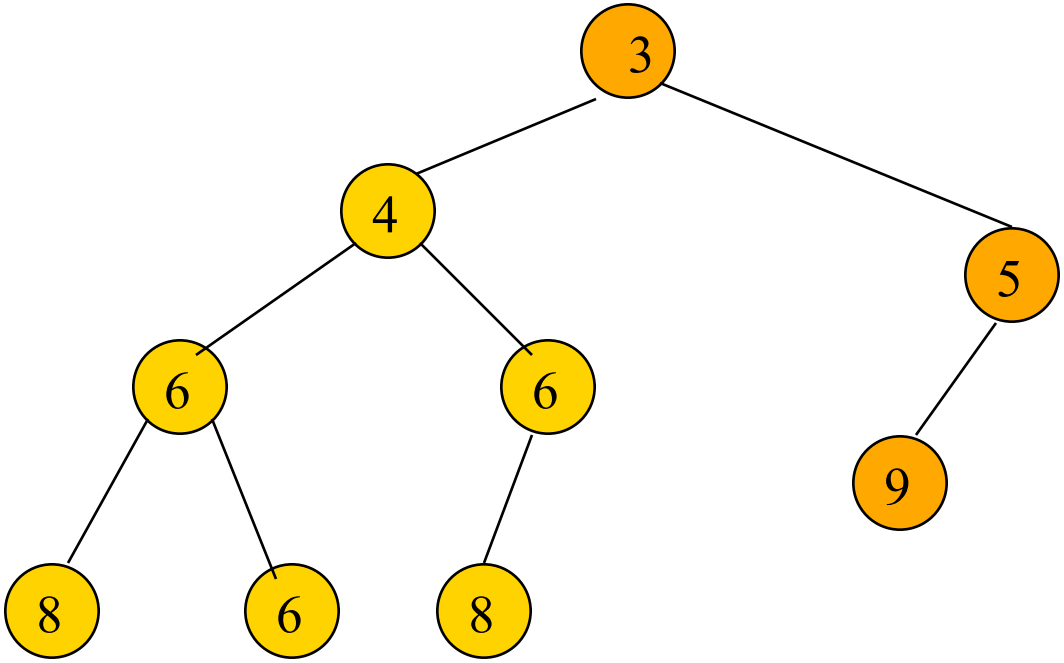
.

( ) < ( ).


3
5
4
9
6
6
8
6
8

.

( ) < ( ).


3
4
5
6
6
9
8
6
8

( )

-

.

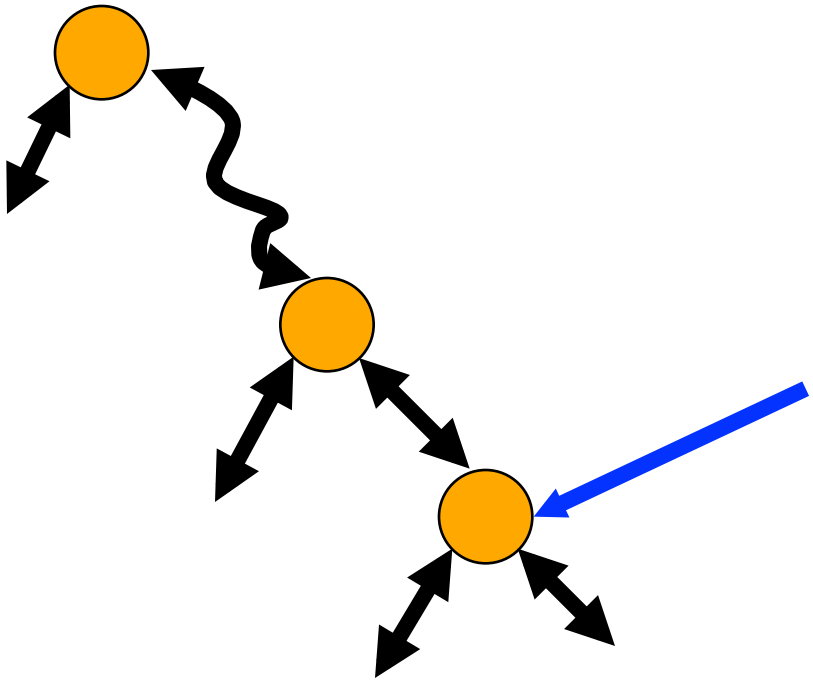
, ,

.

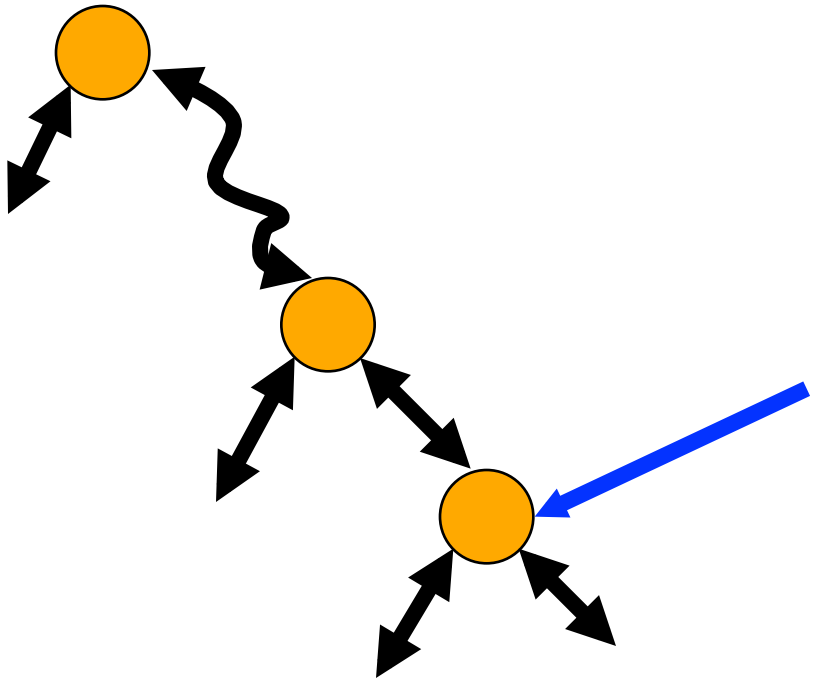
1

.

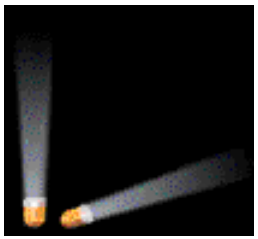
.

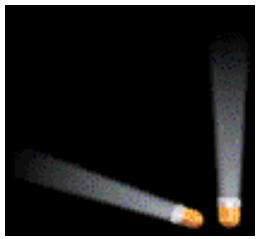


, !=



/





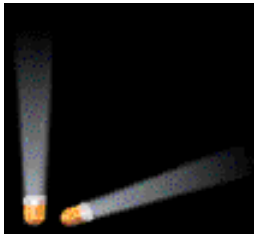
( )

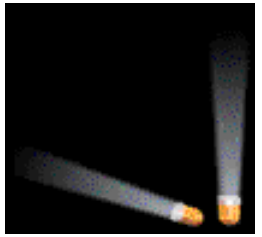
: 2 →

2 :

: →

:





16

8 1/8 → 8

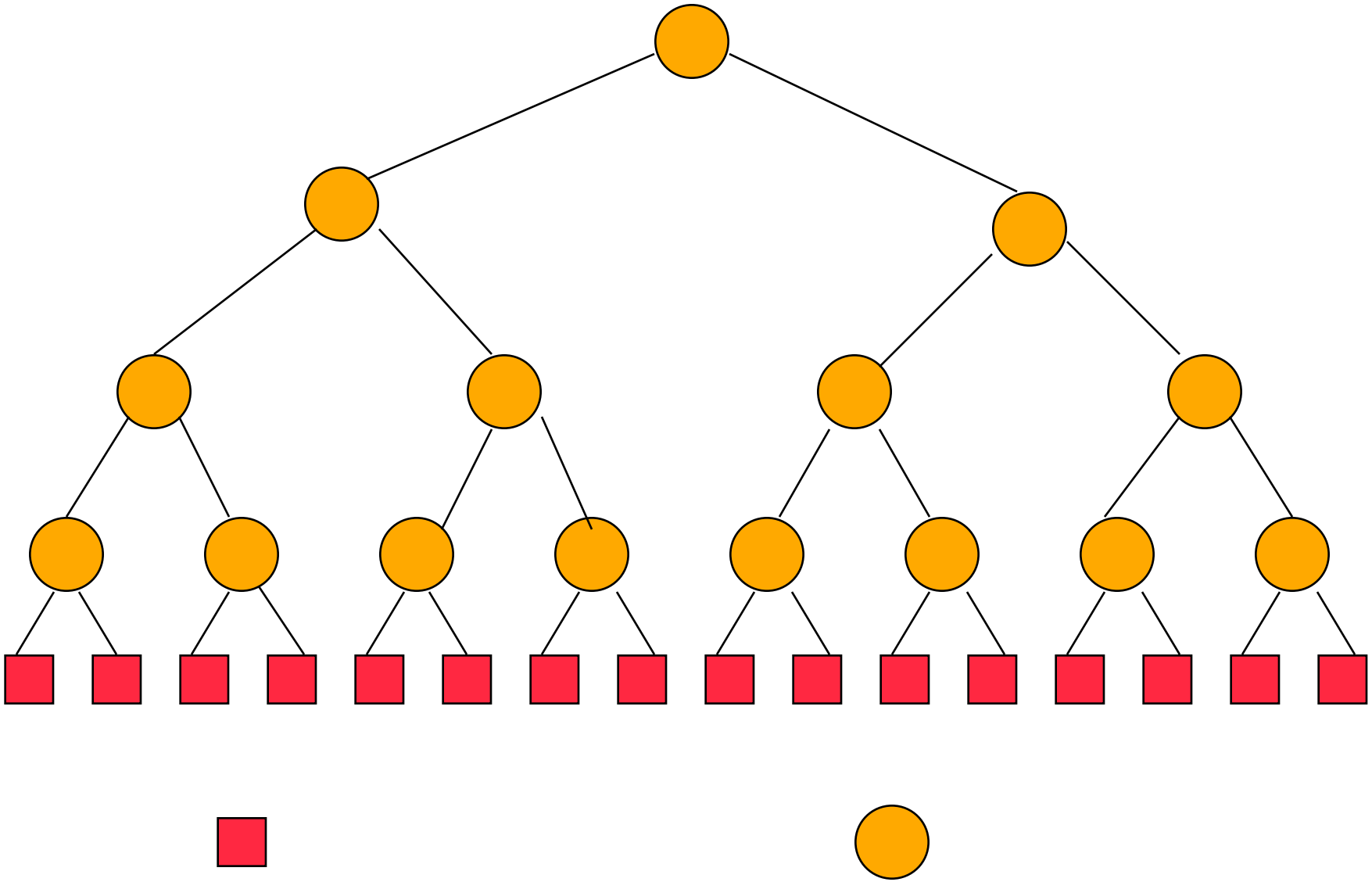
4 1/4 → 4

2 → 2

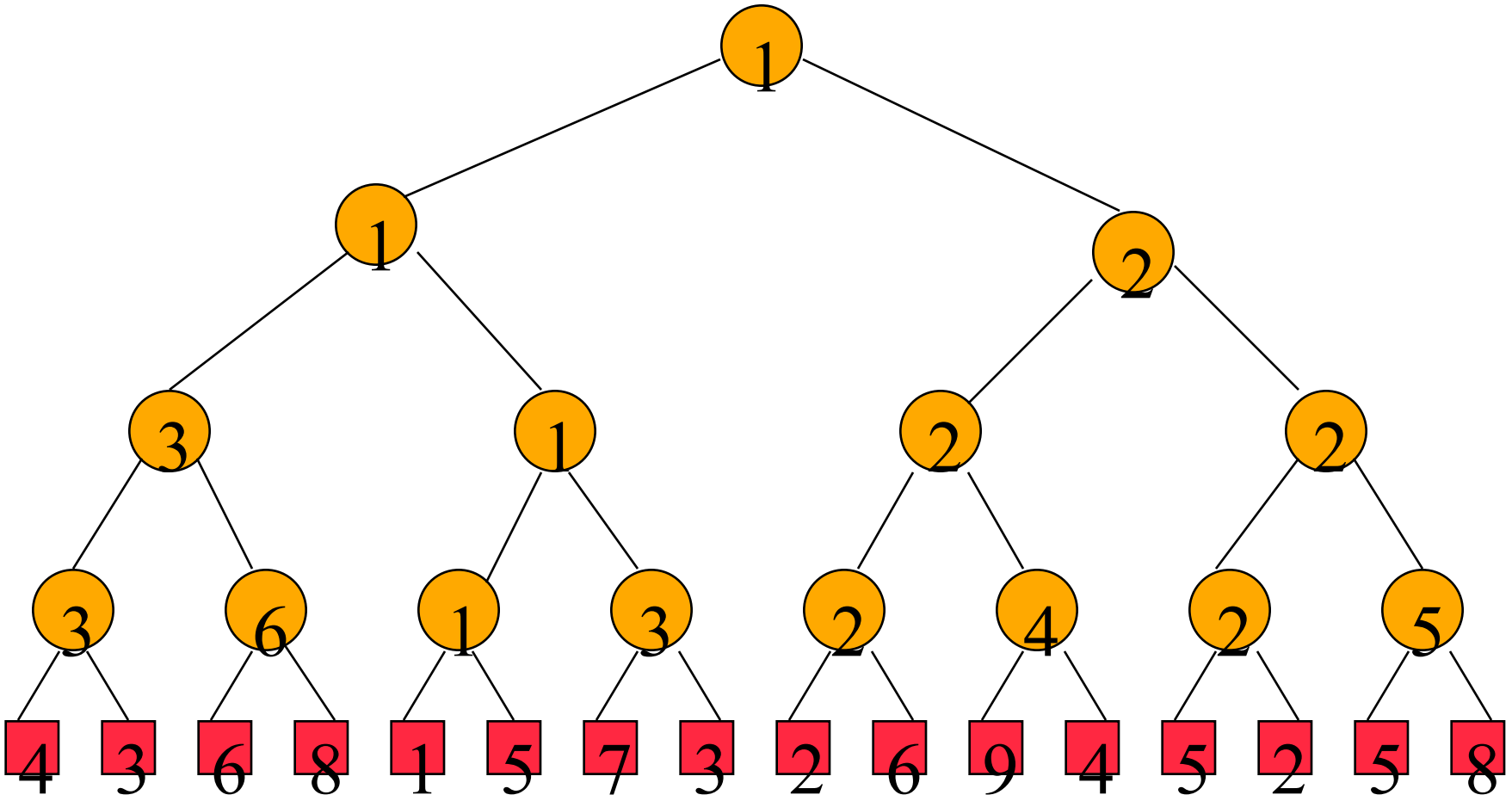
1 →

:

16

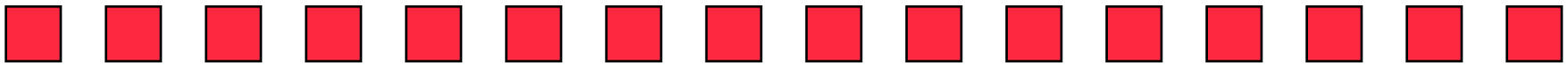


16



$\Rightarrow$.

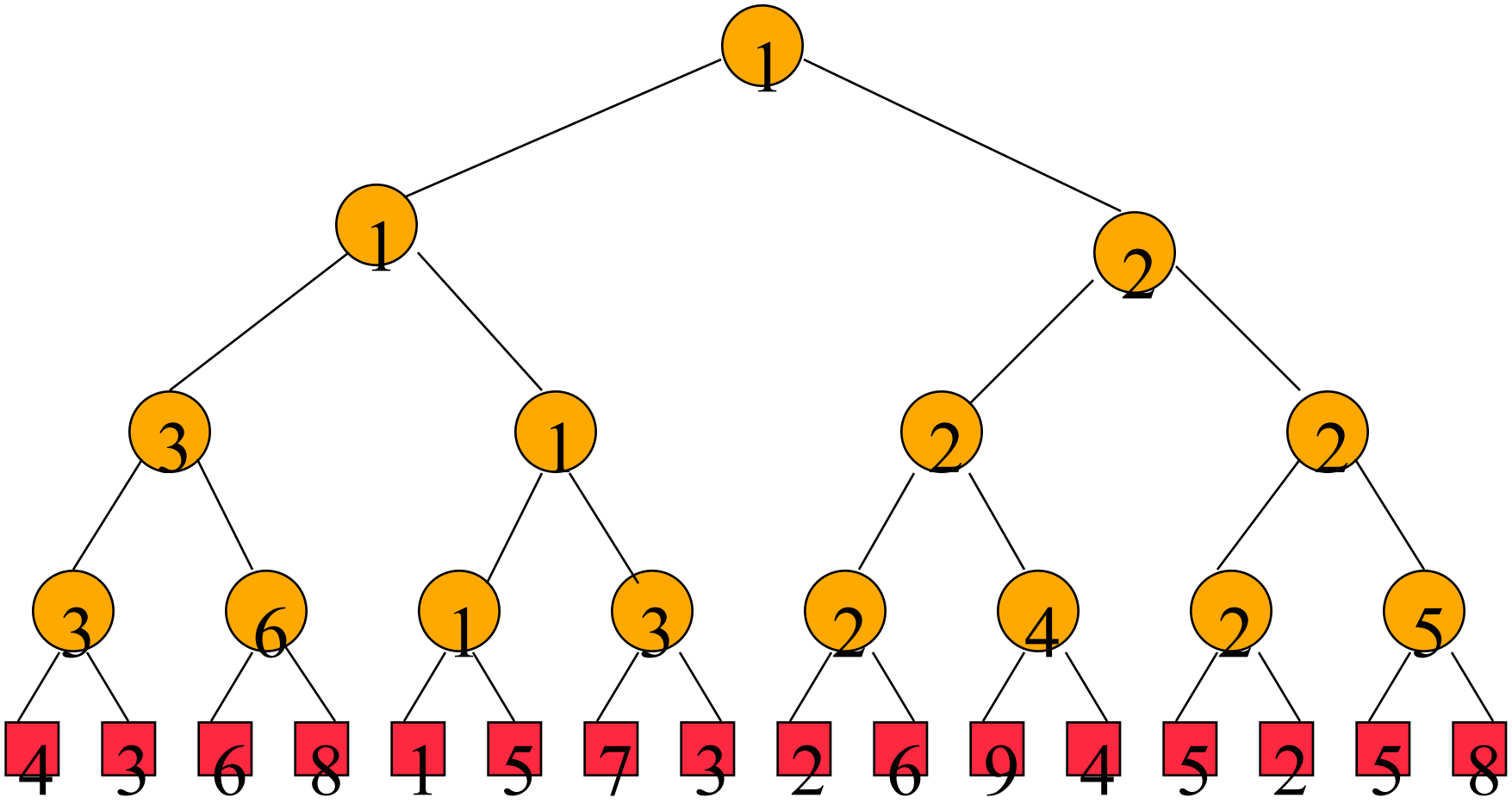
16



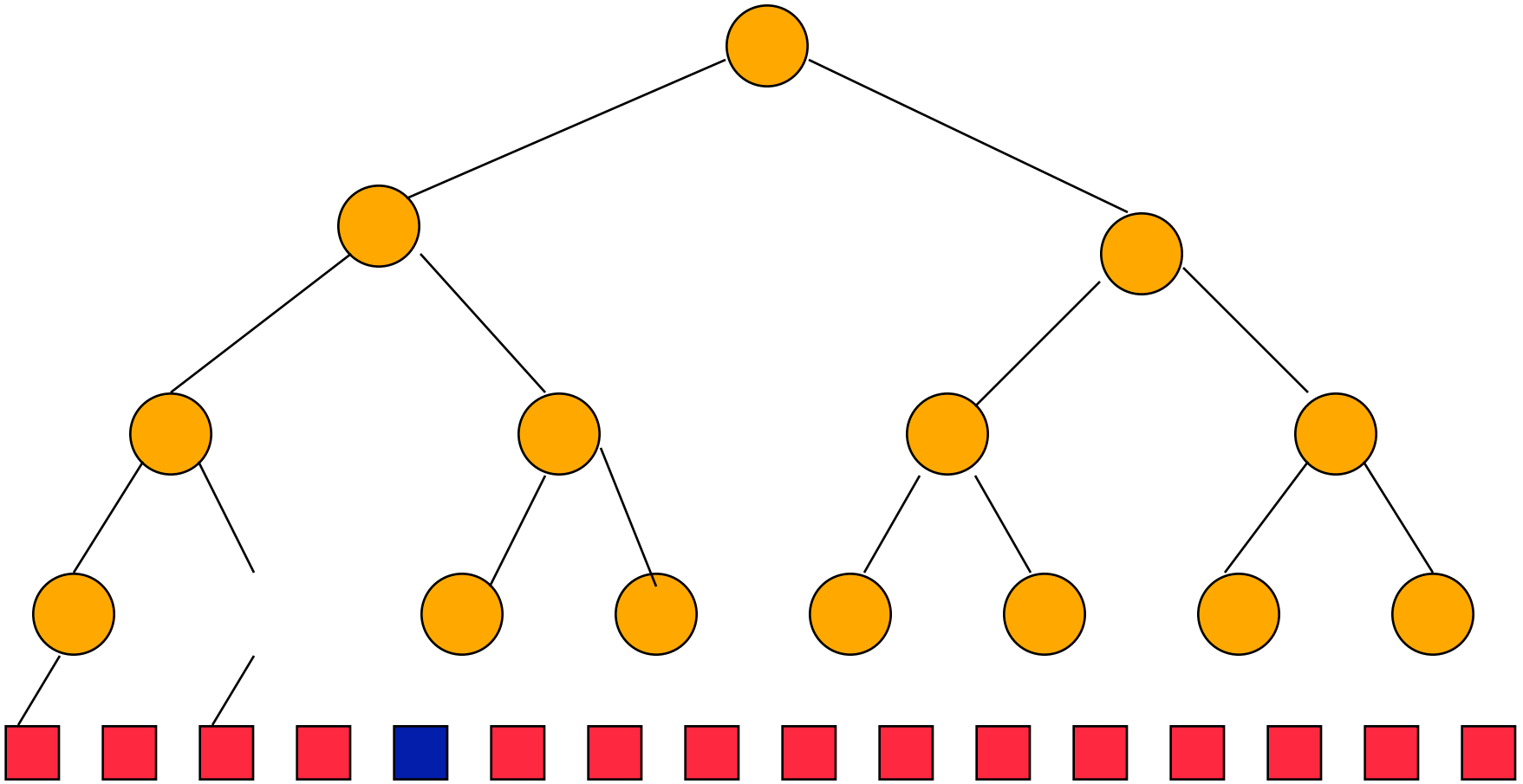
(1)

- 1

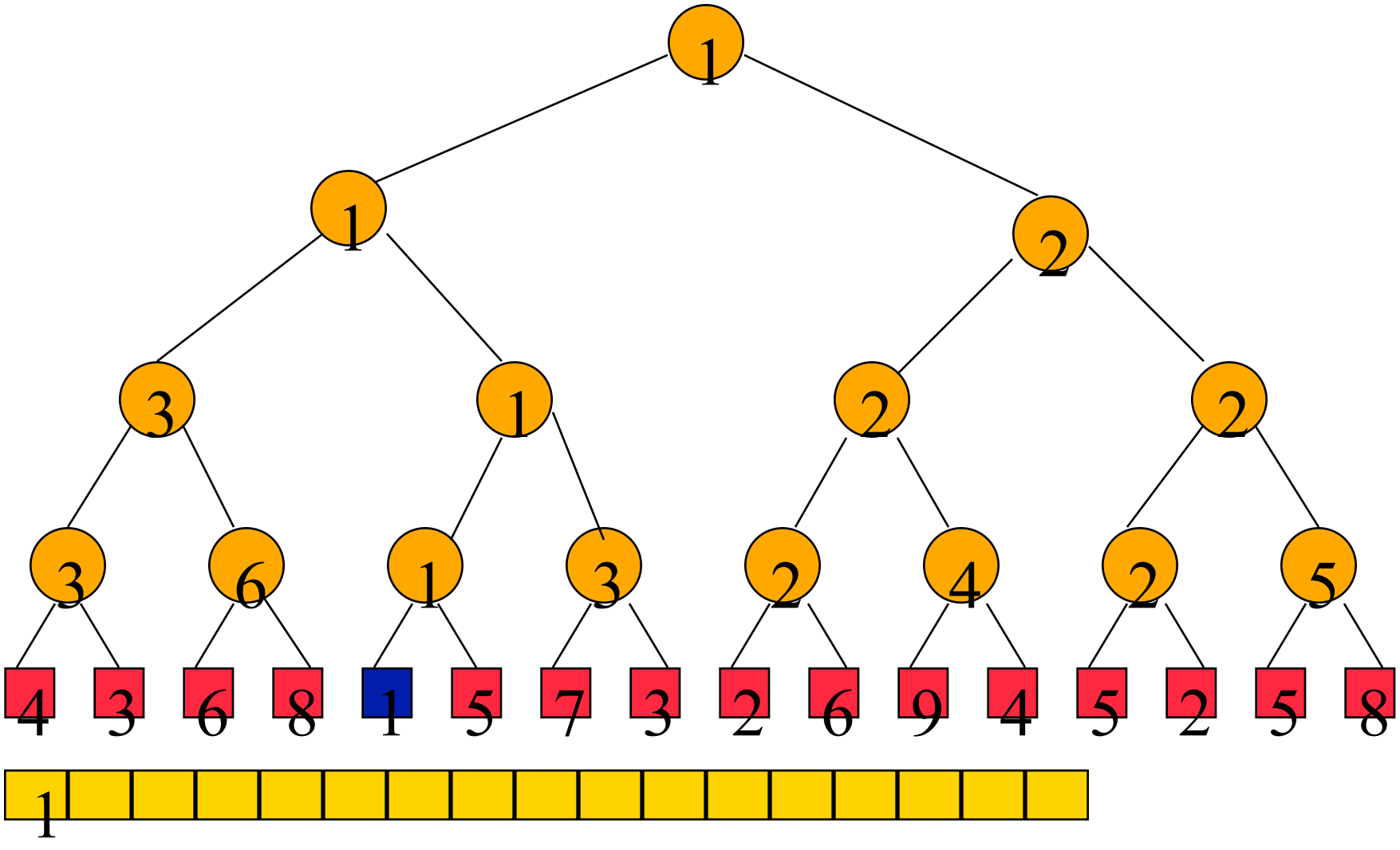
( )


1
1
2
3
1
2
2
3
6
1
3
2
4
2
5
4
3
6
8
1
5
7
3
2
6
9
4
5
2
5
8

16

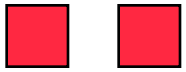


16

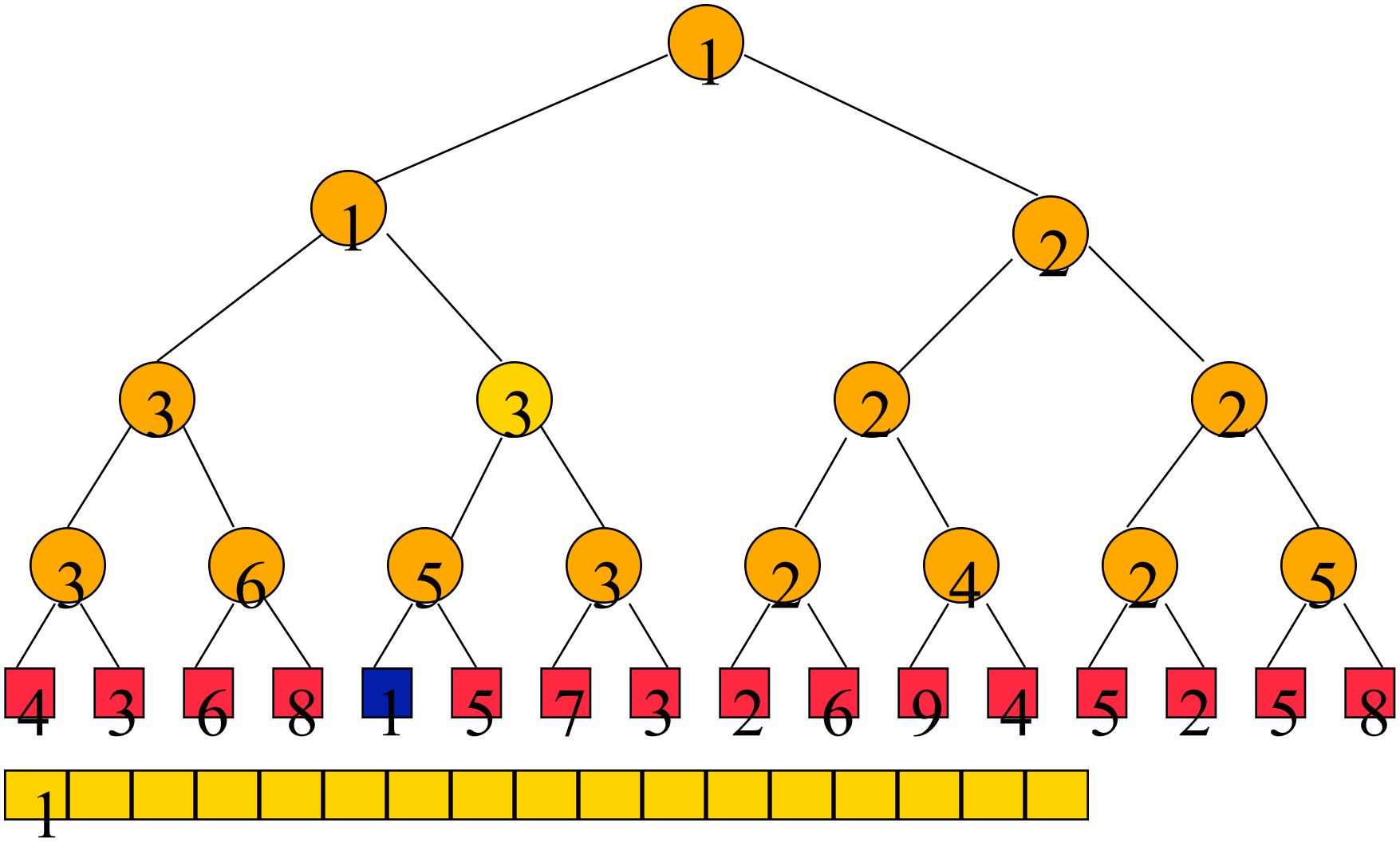


.

16

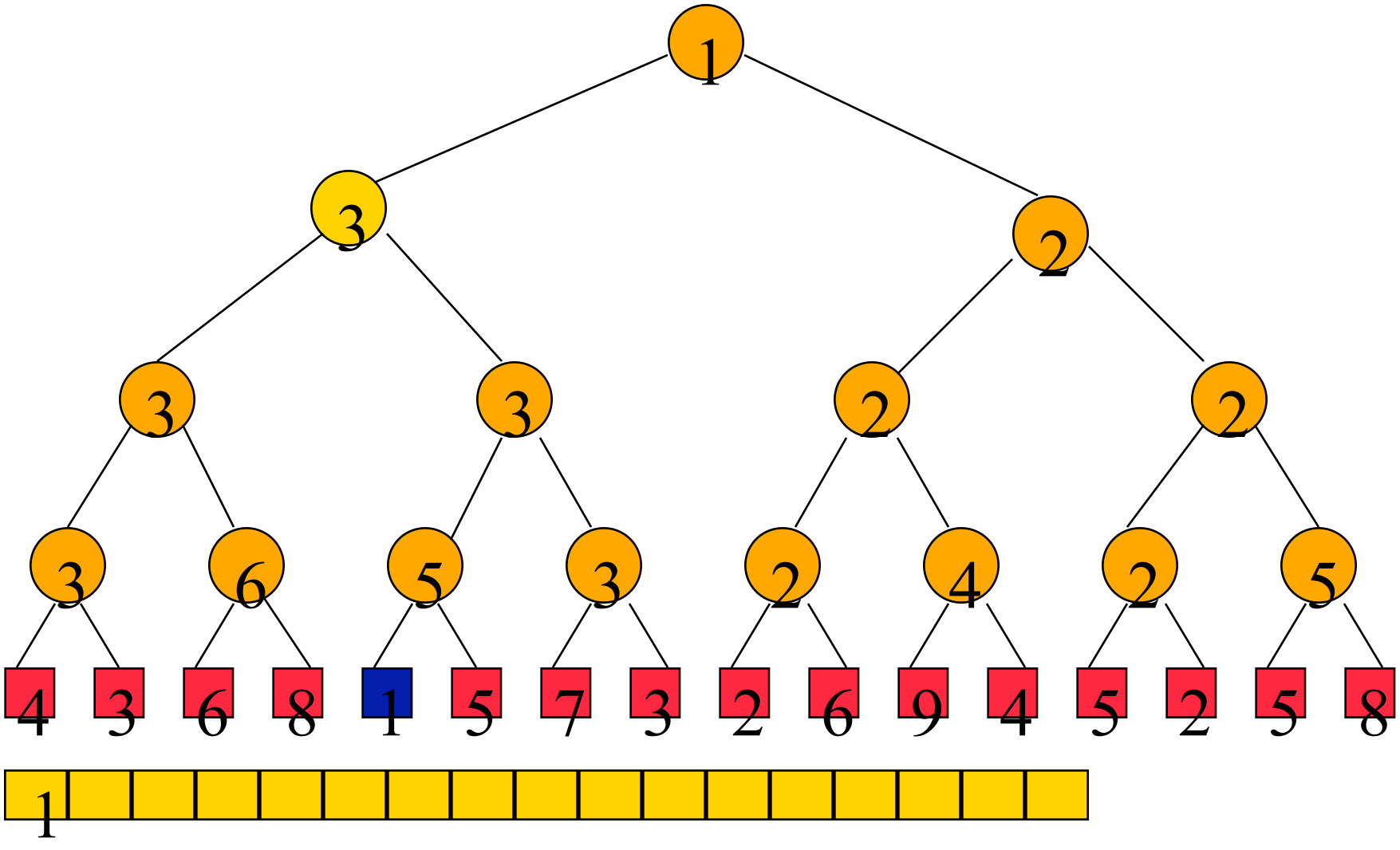


16



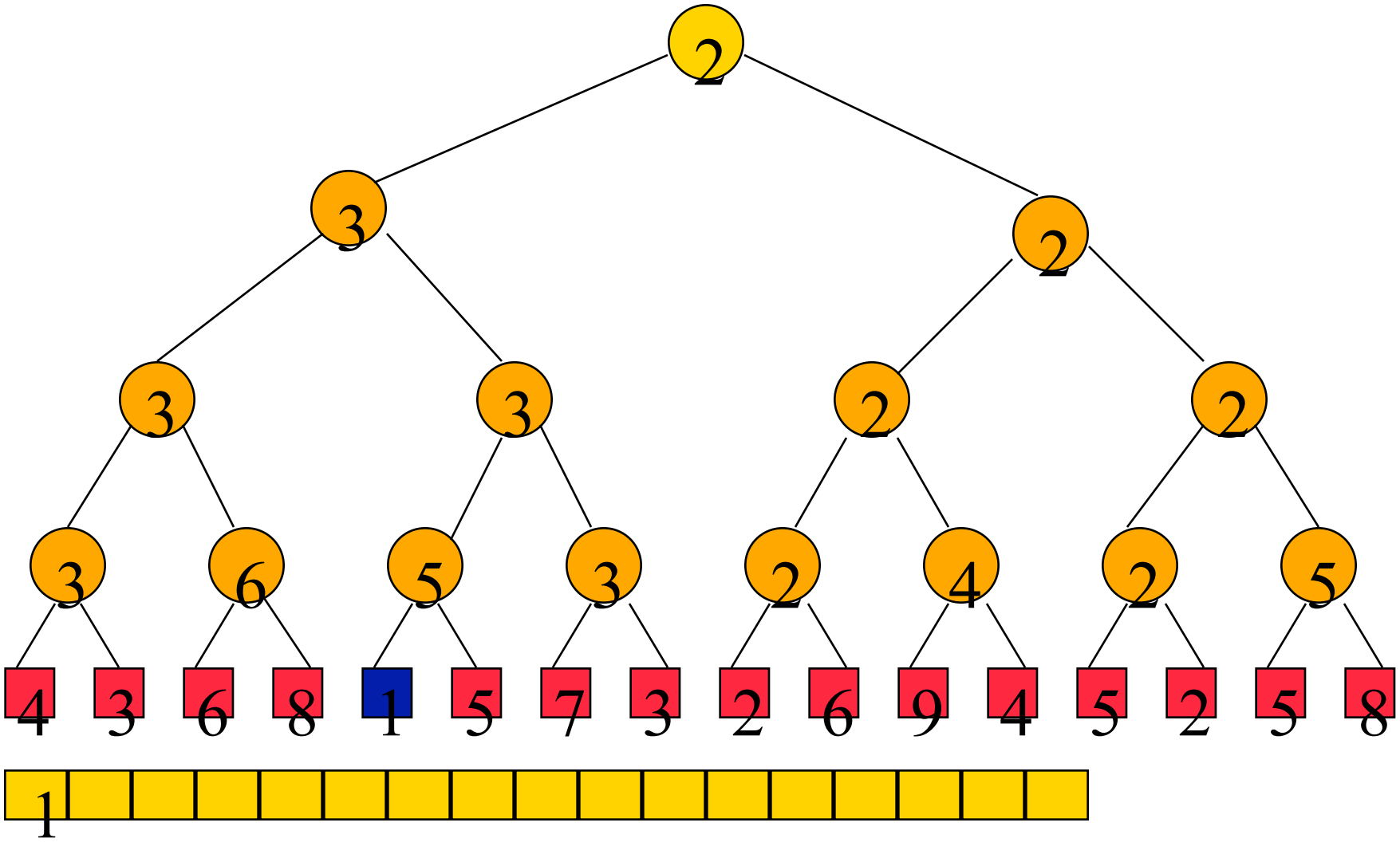
.

16



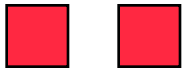
.

16

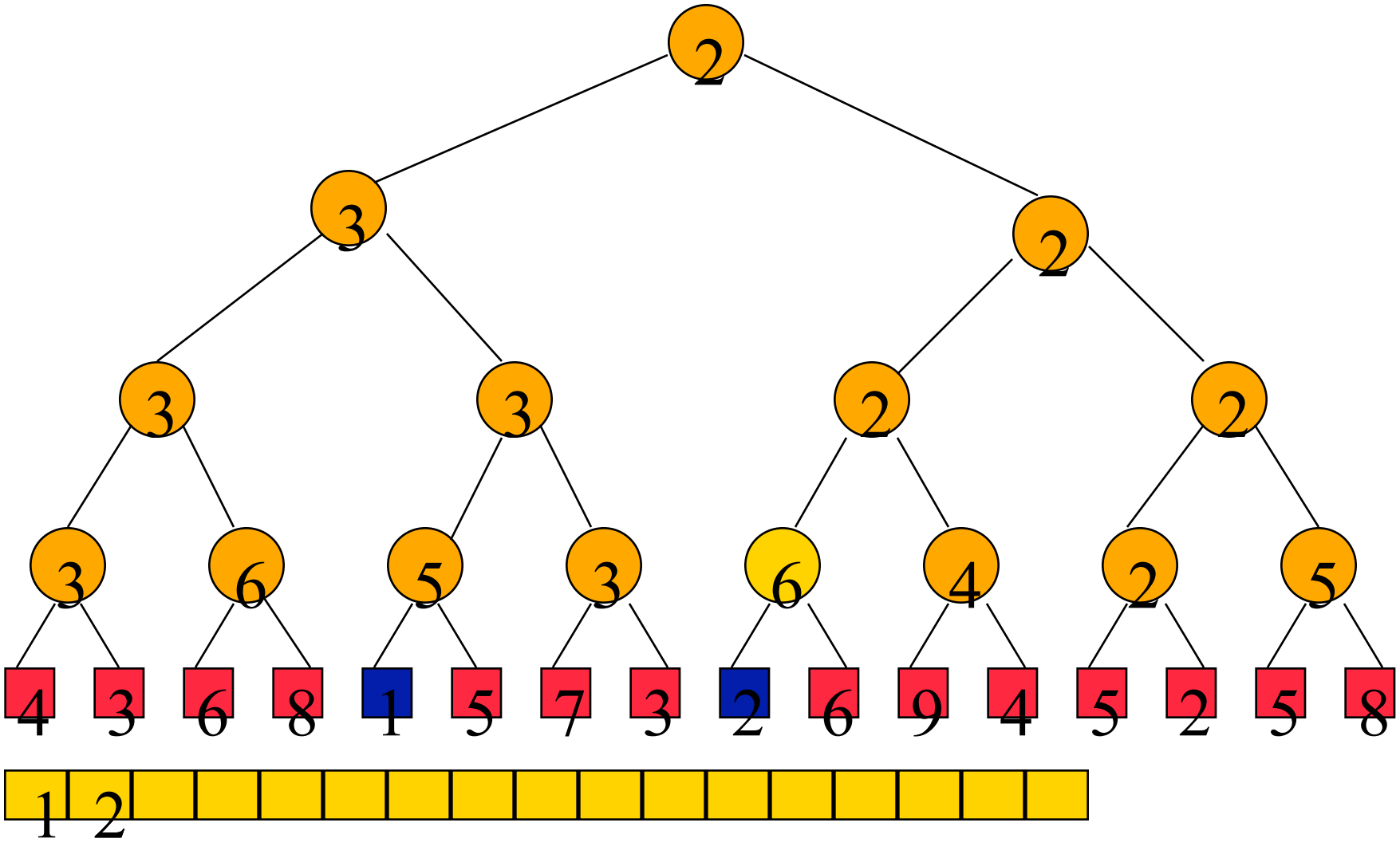


.

16

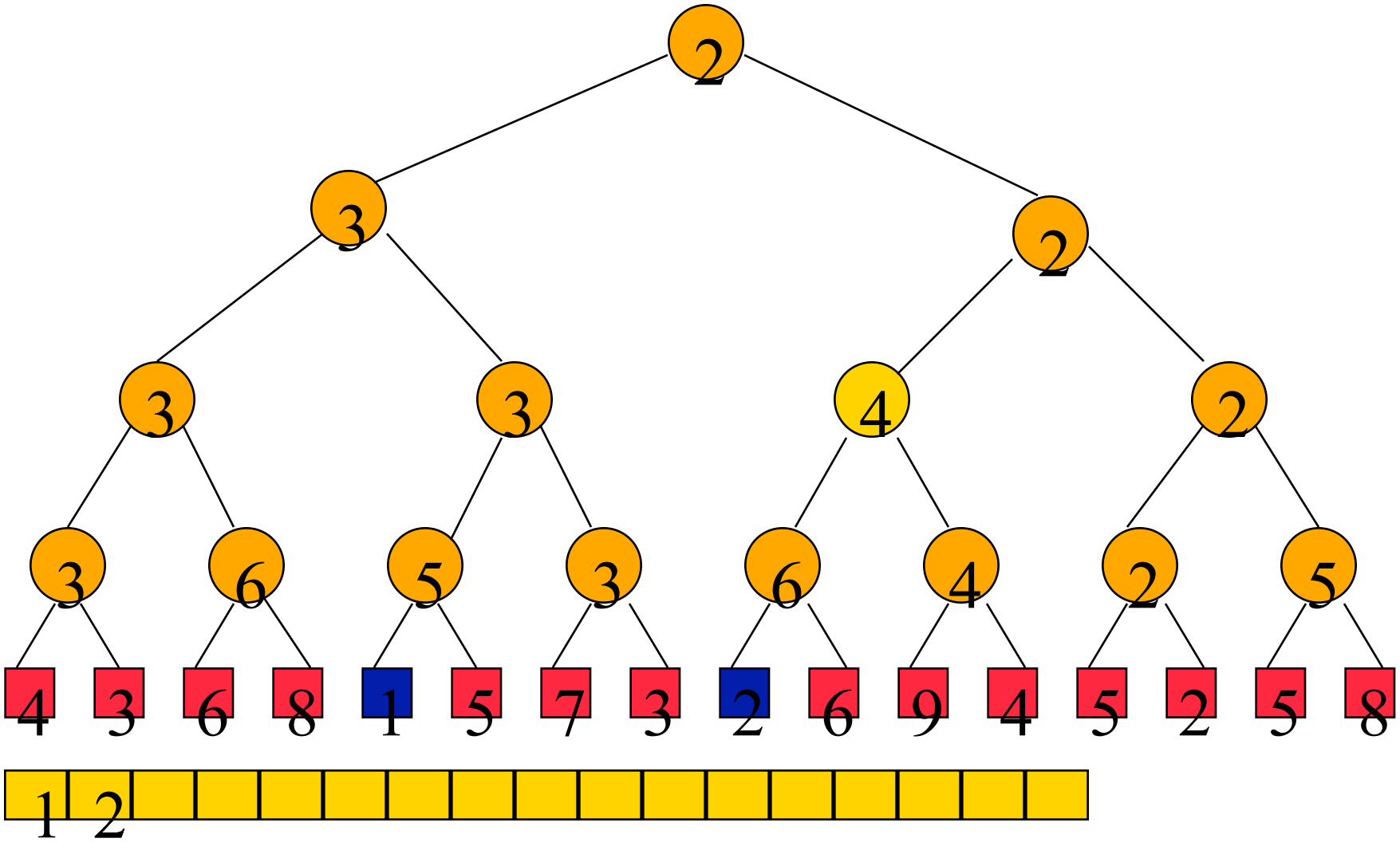


16



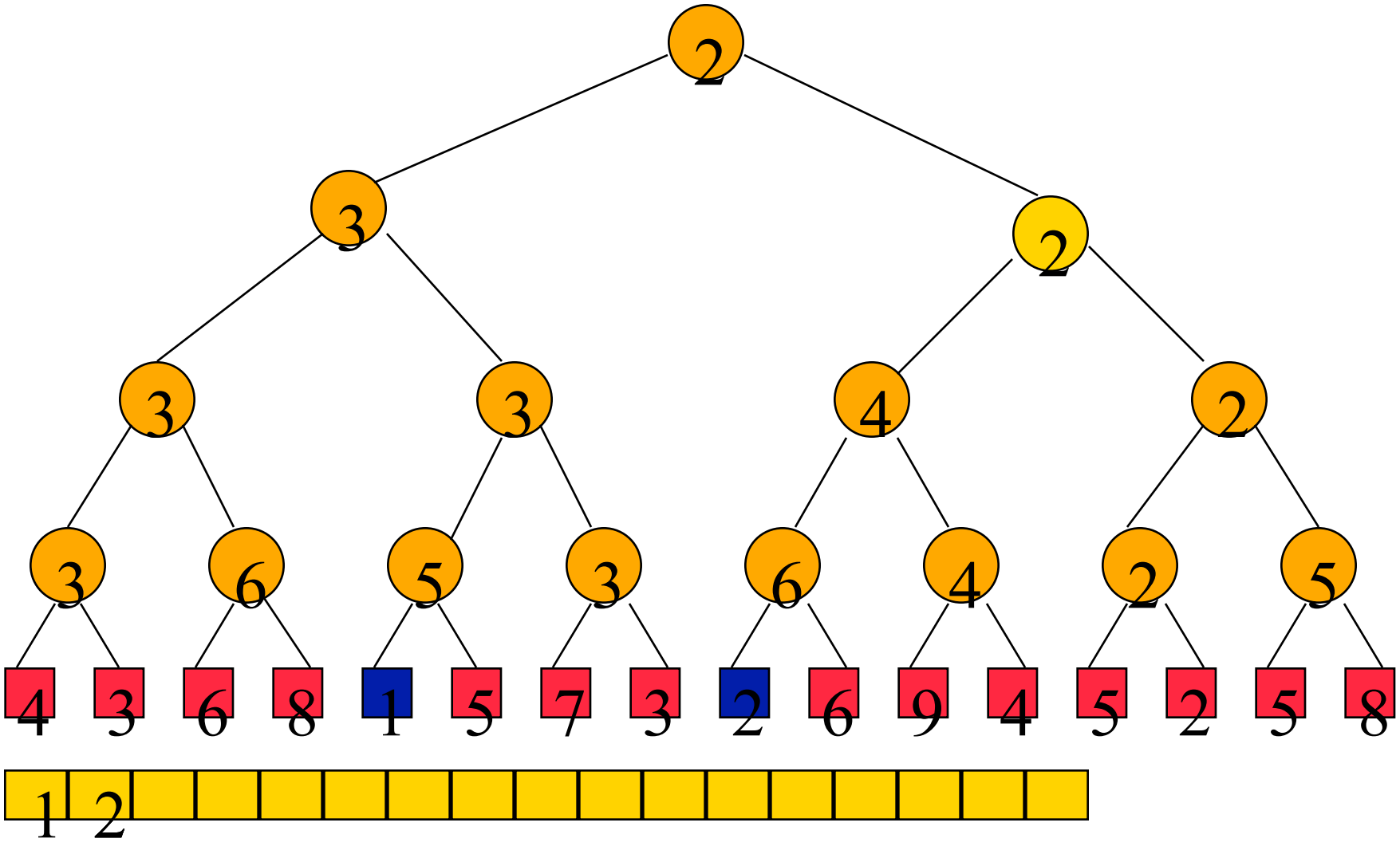
.

16



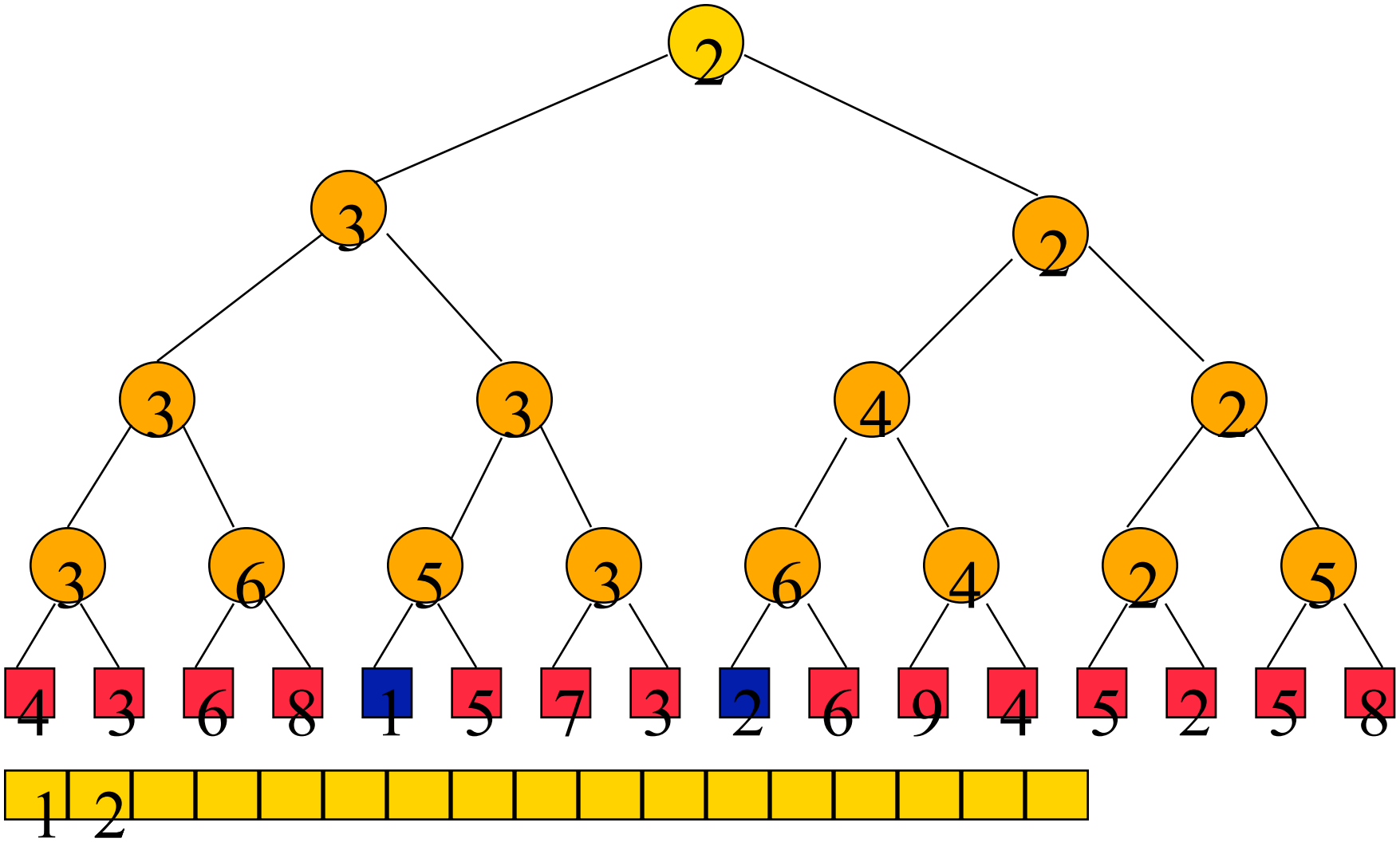
.

16



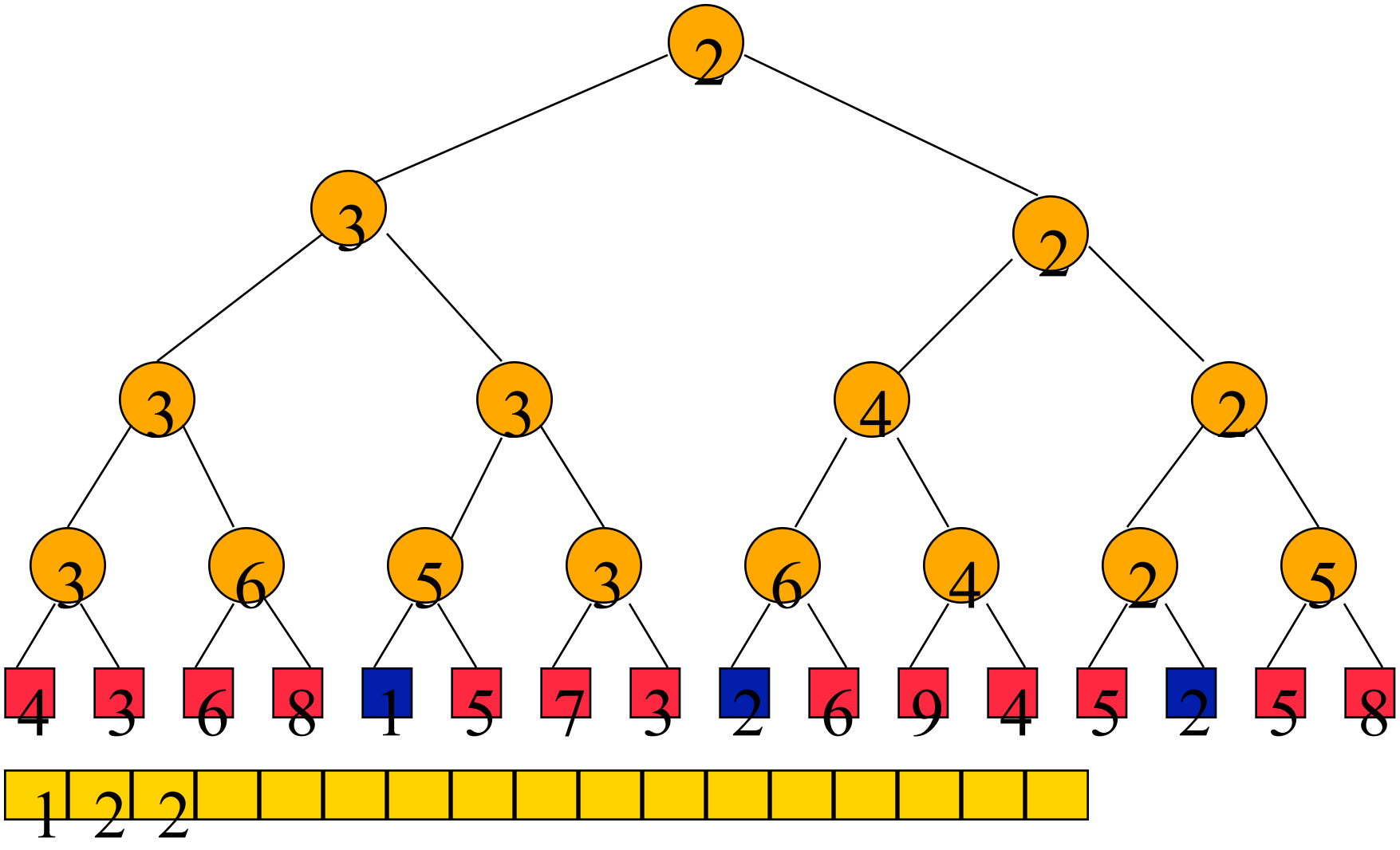
.

16



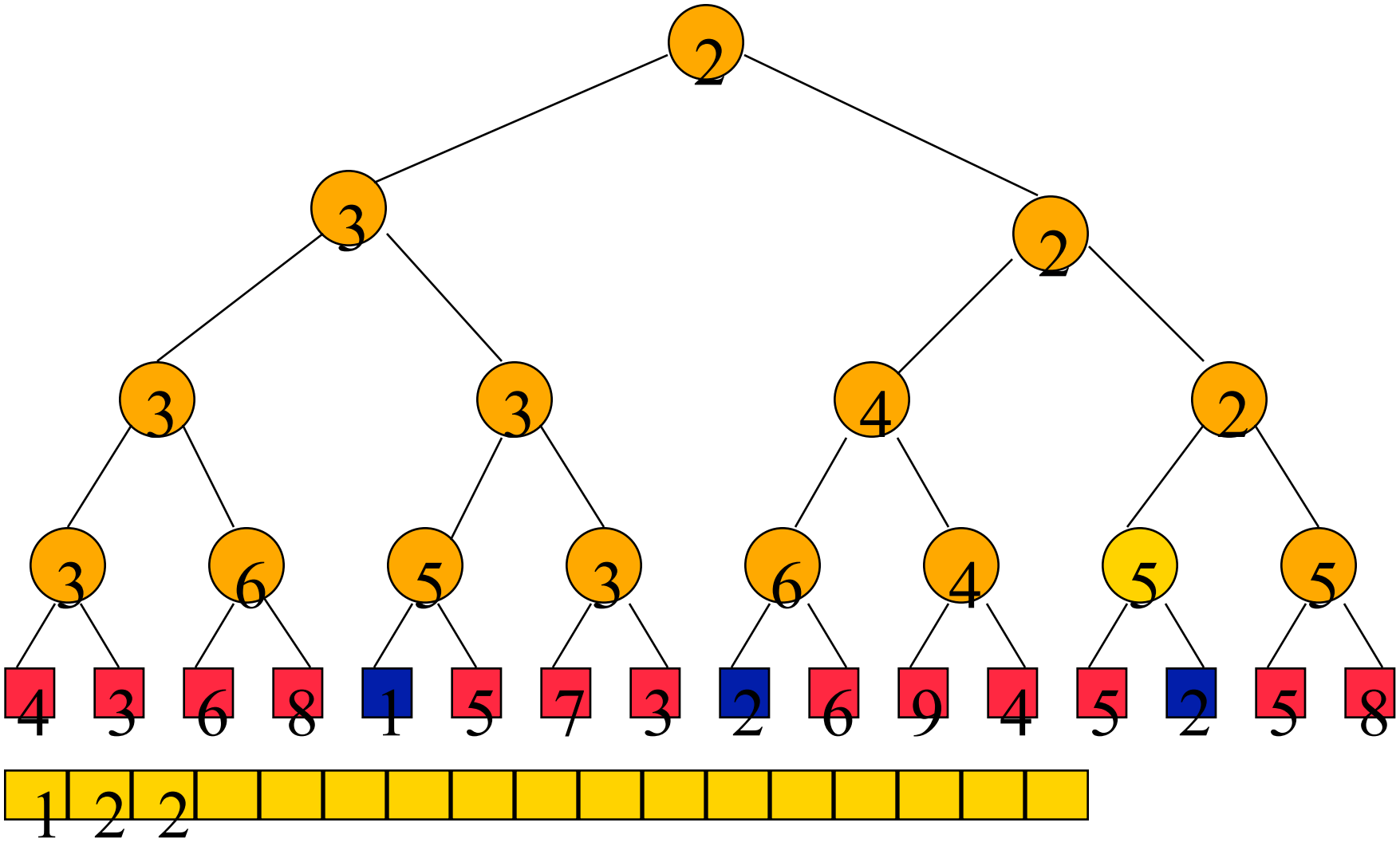
.

16



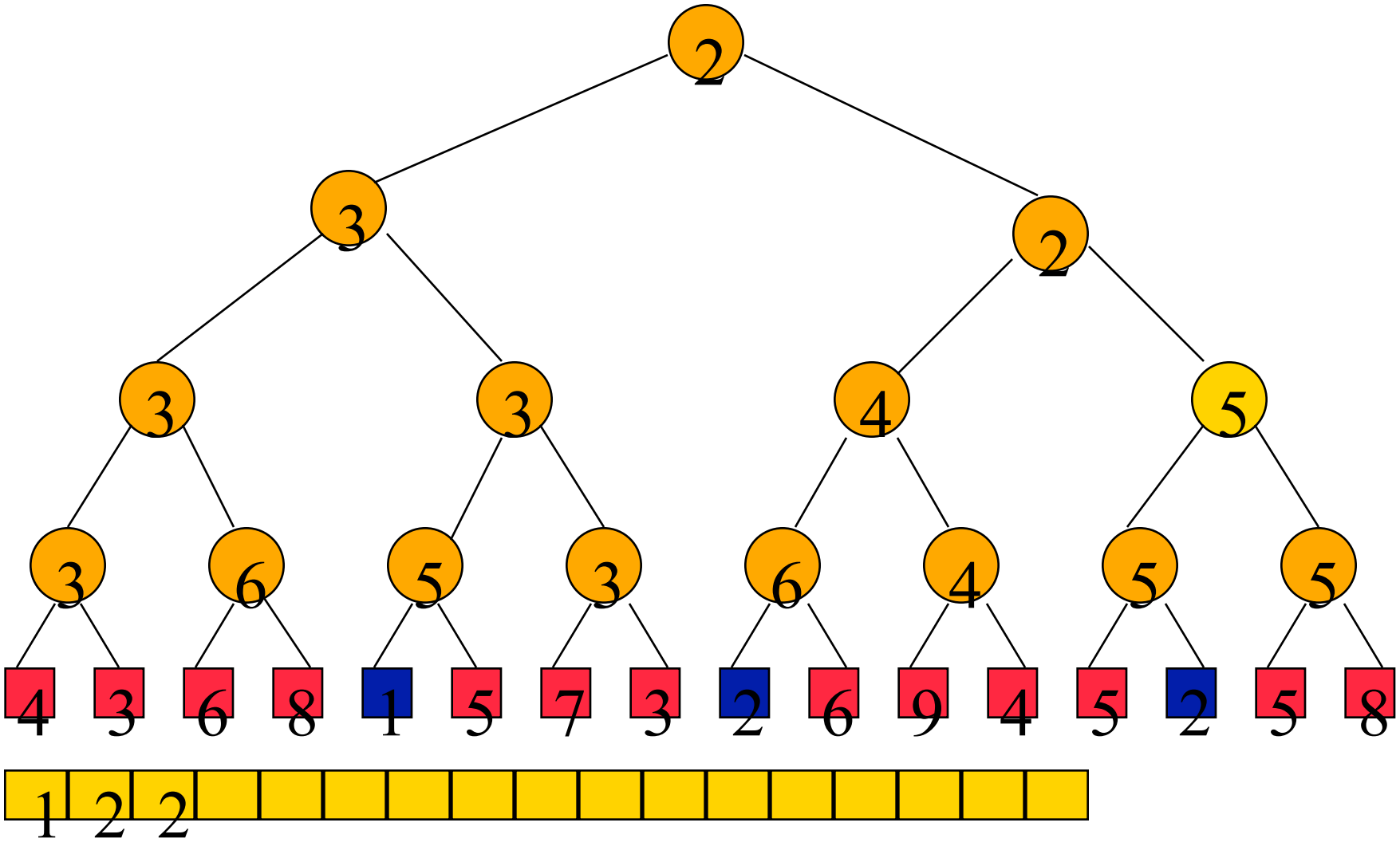
.

16



.

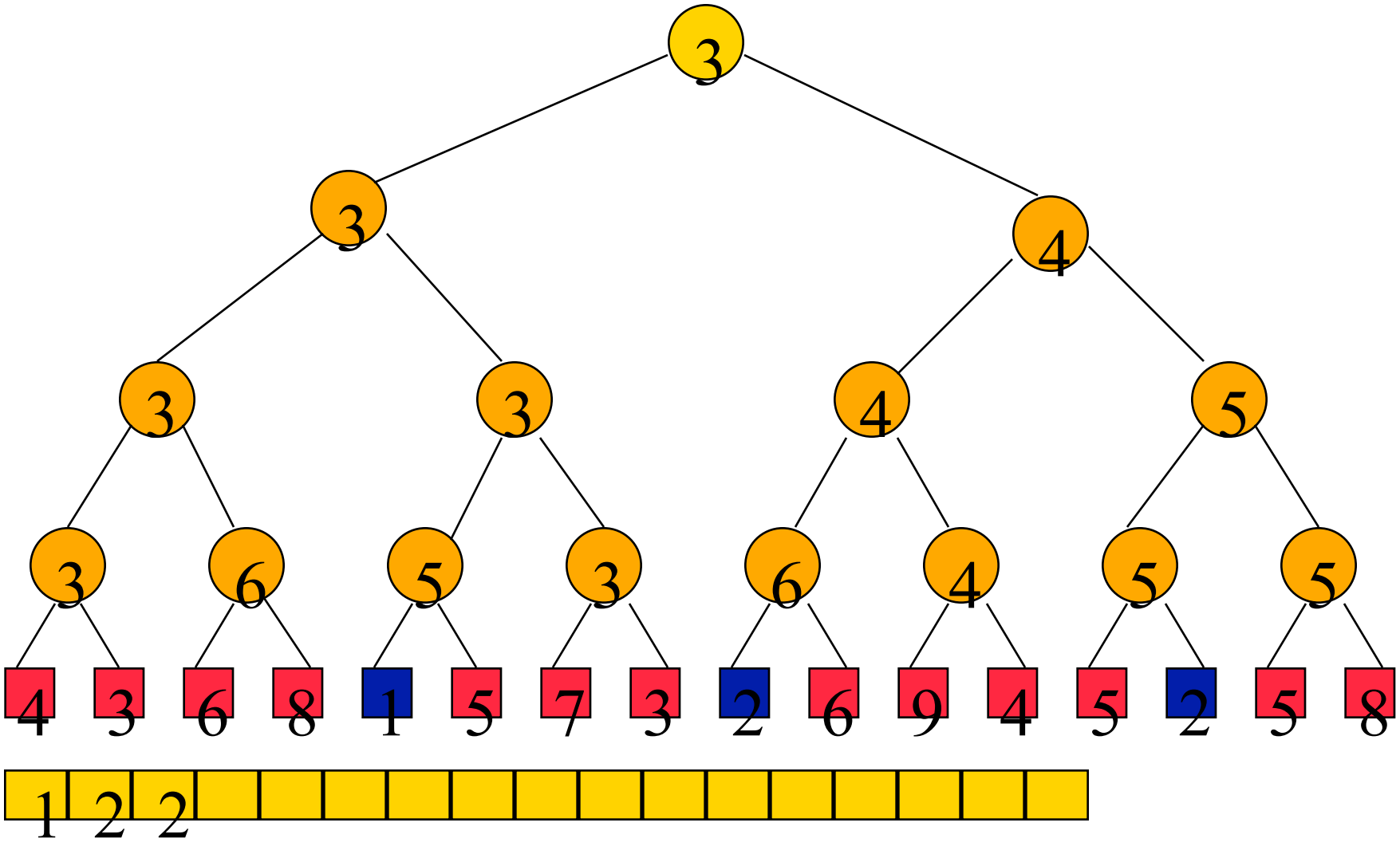
16



.

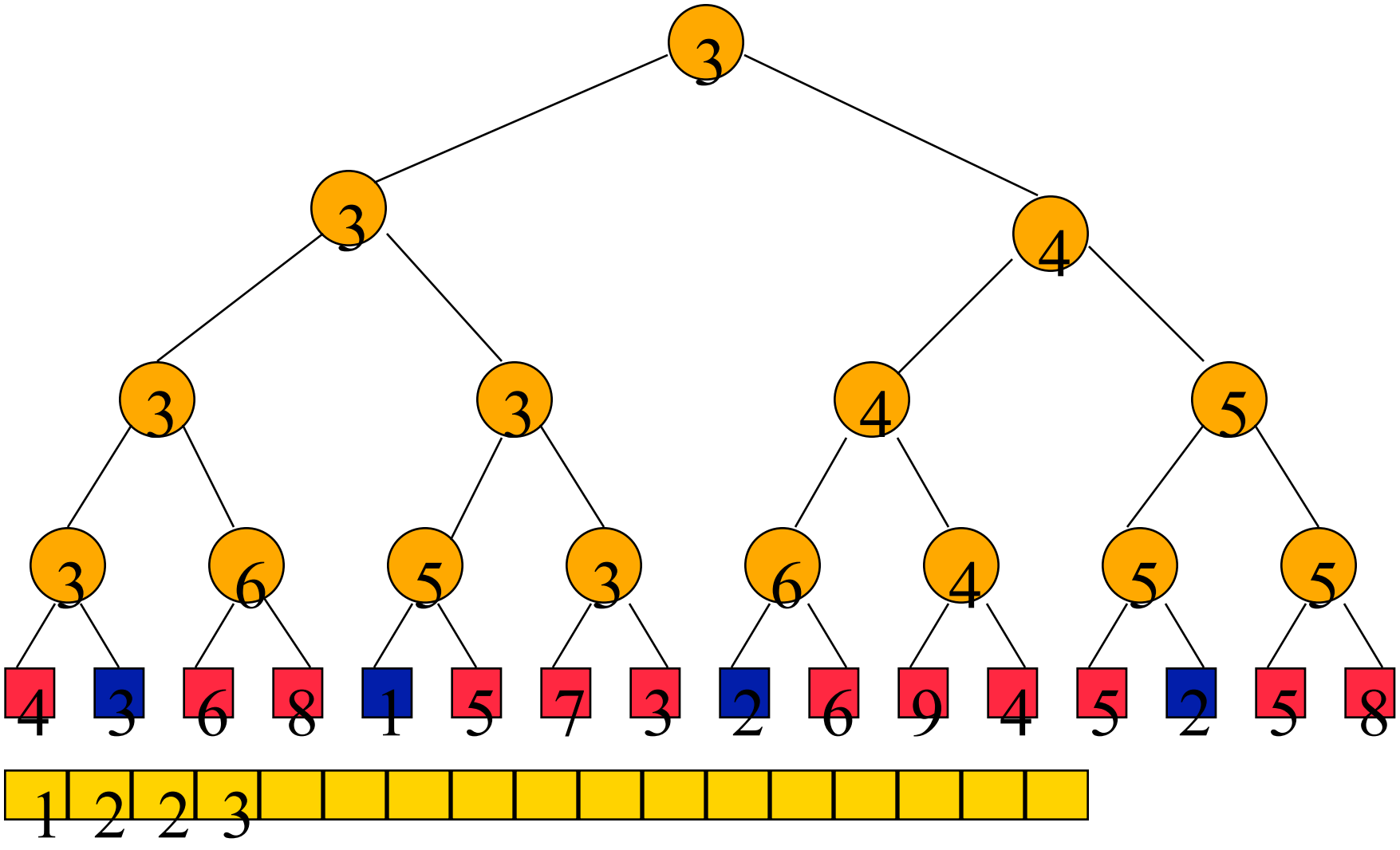
16

16



.

16

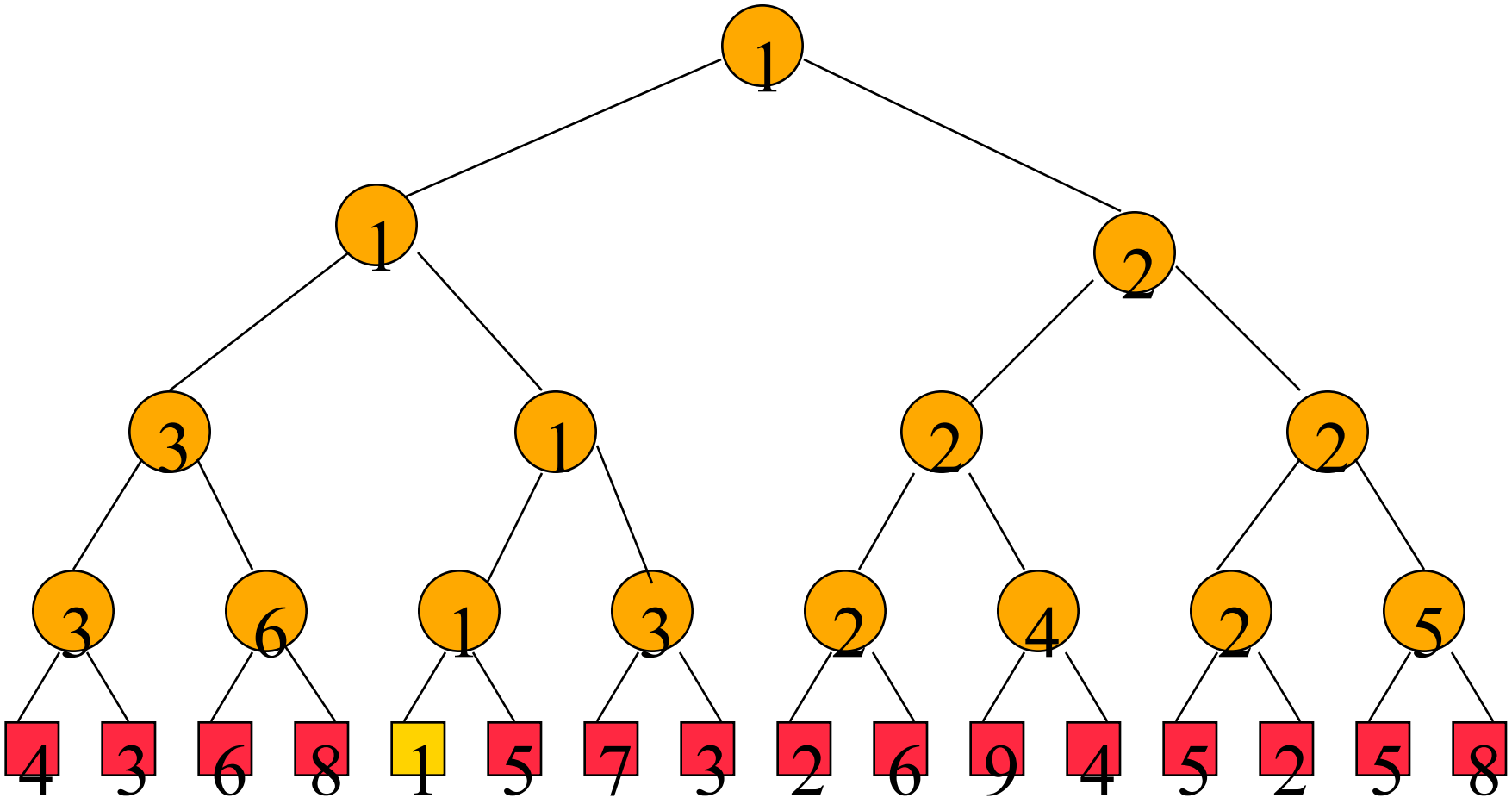


.

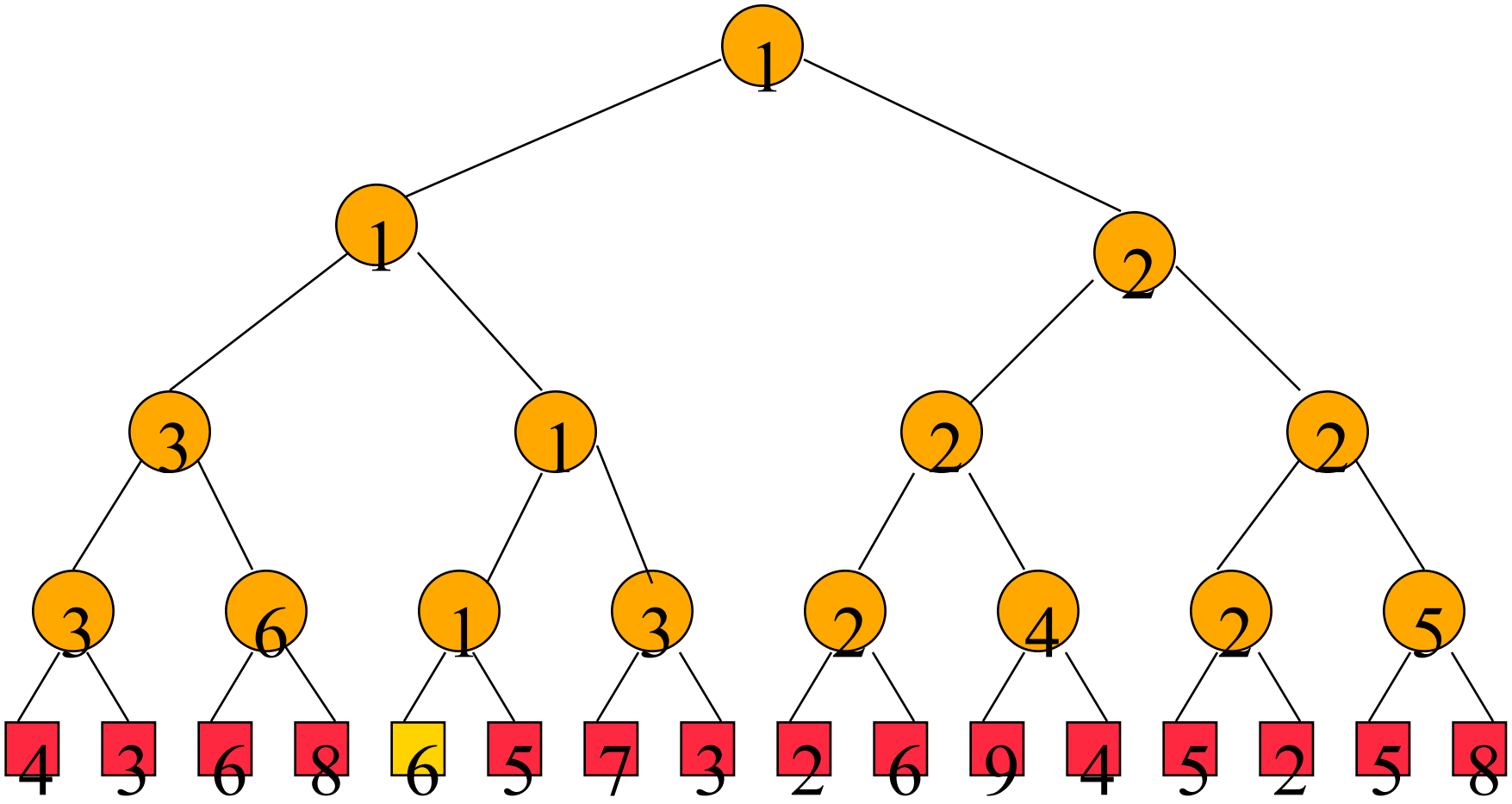
- ( )
- ( )
- ( )

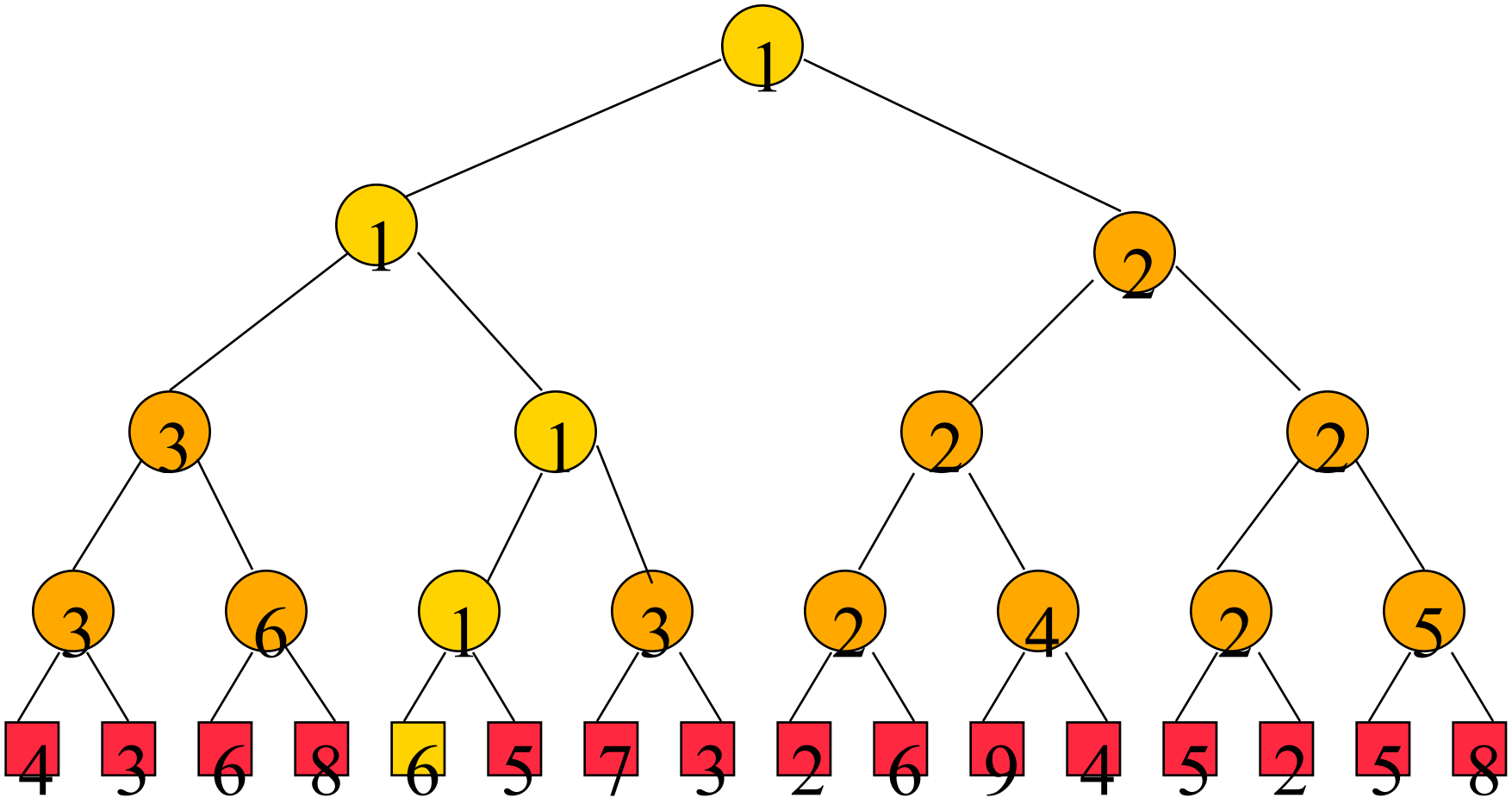
( ).

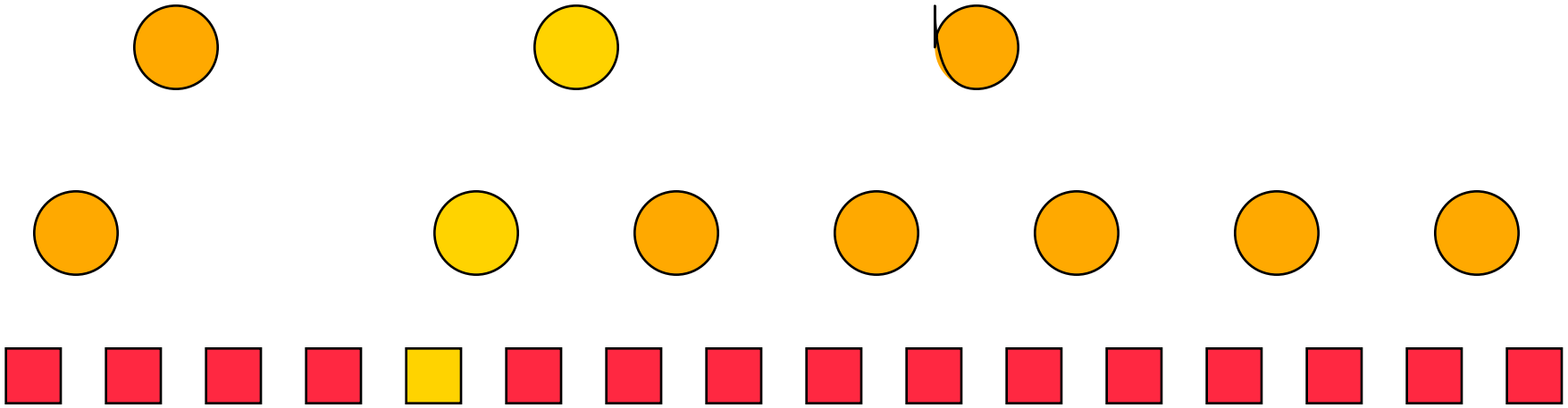
- ( )
- (1)
  /
- ( )
- ( )

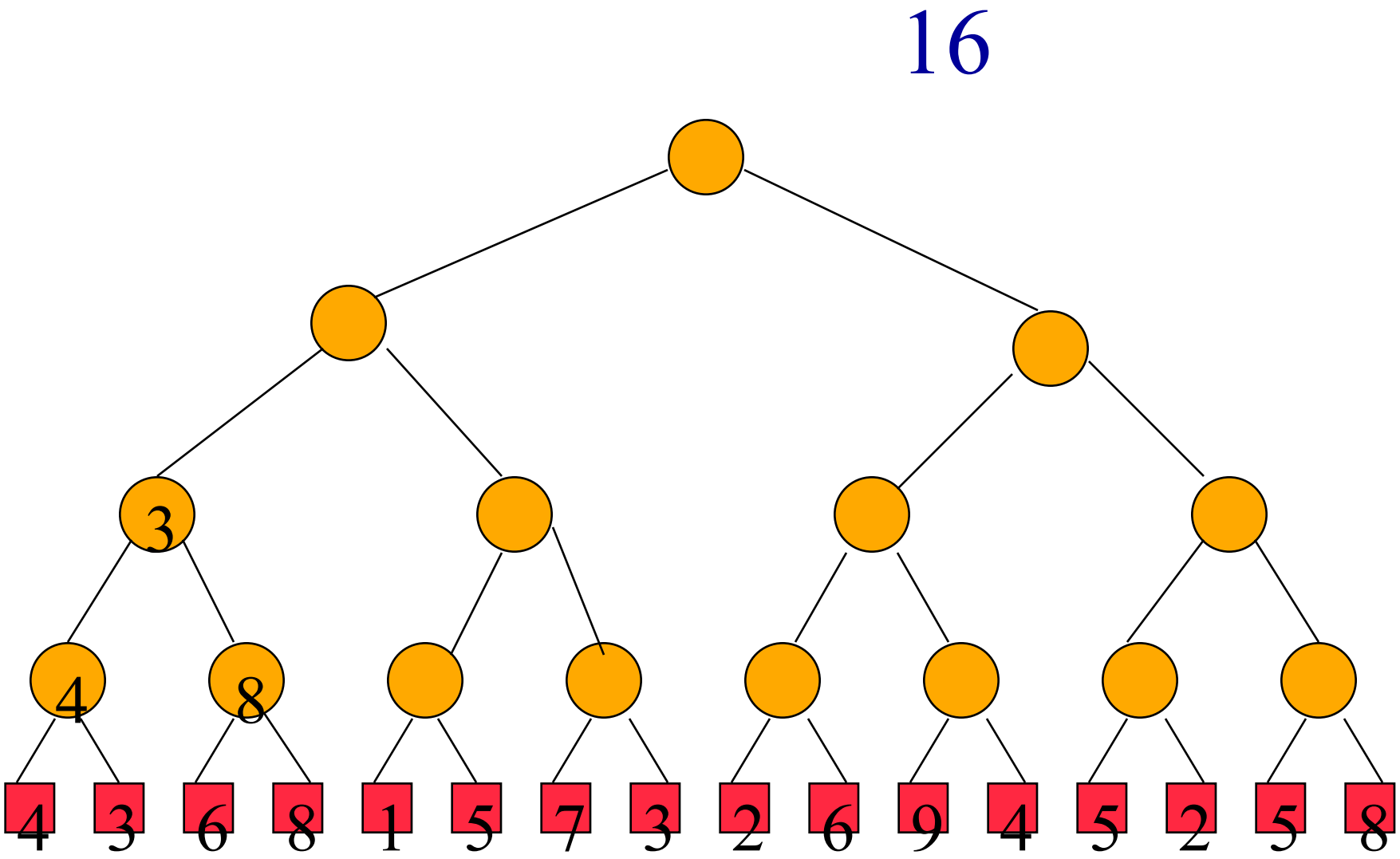


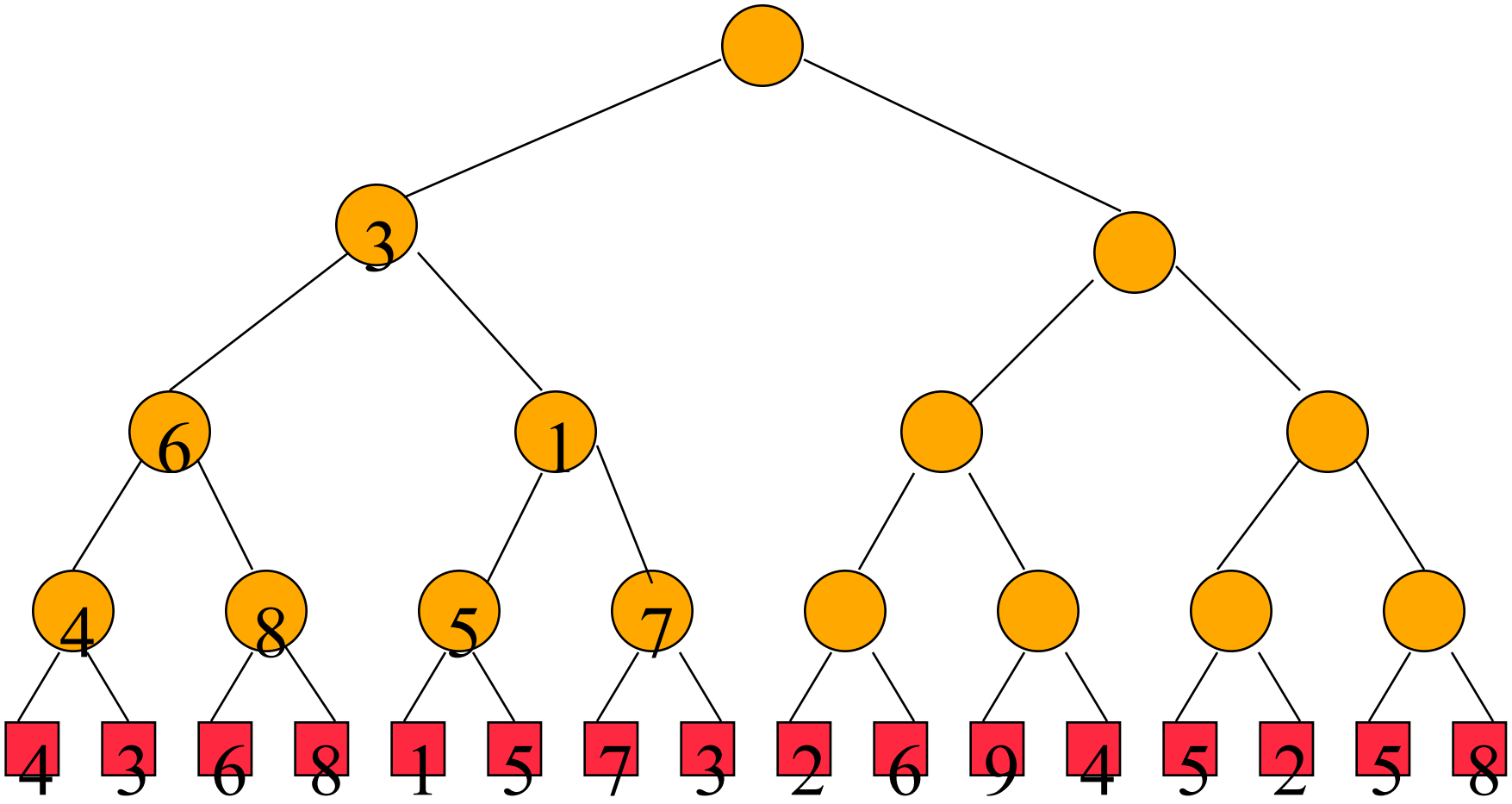
6.

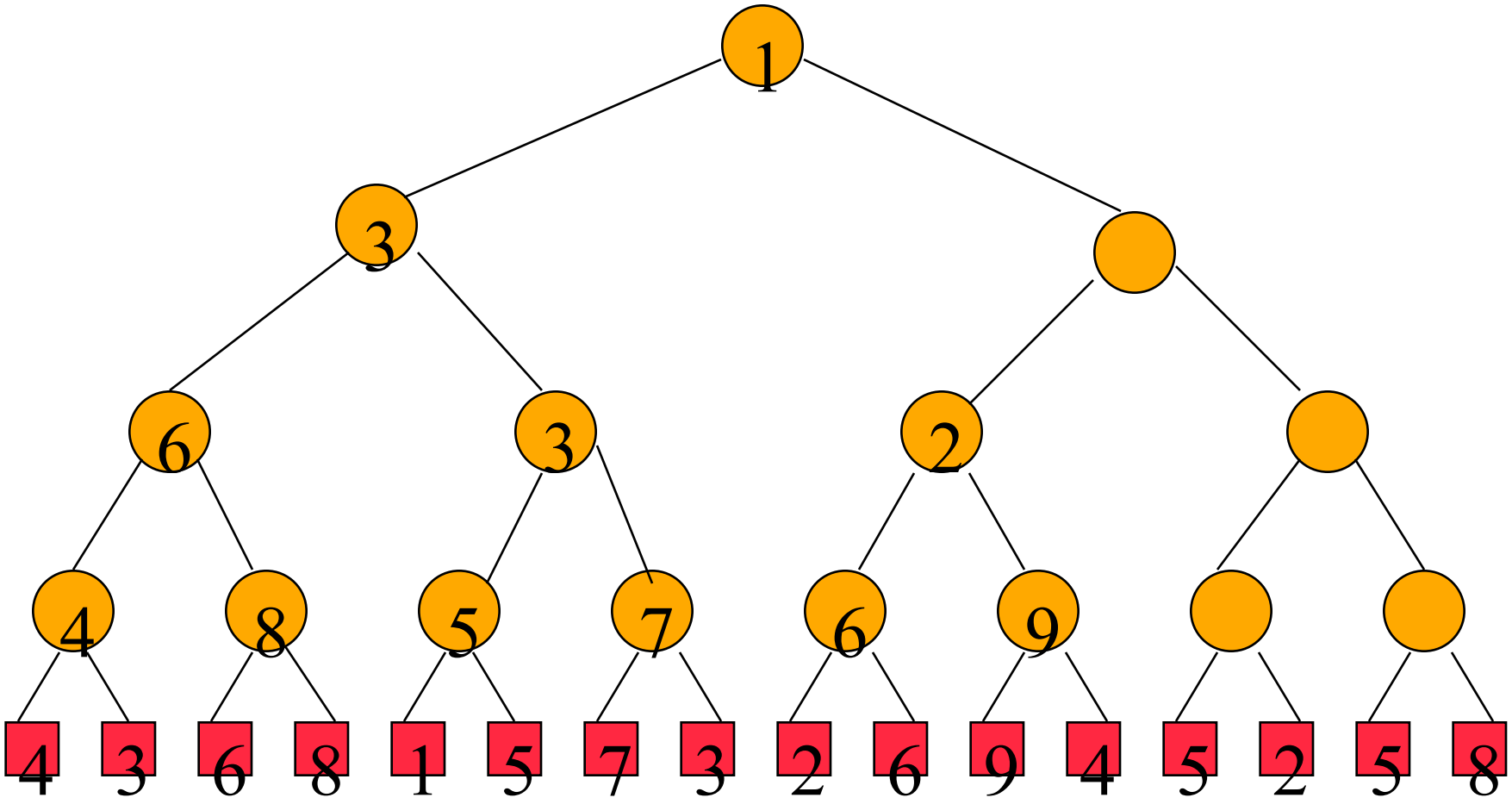

1
1
2
3
1
2
2
3
6
1
3
2
4
2
5
4
3
6
8
6
5
7
3
2
6
9
4
5
2
5
8

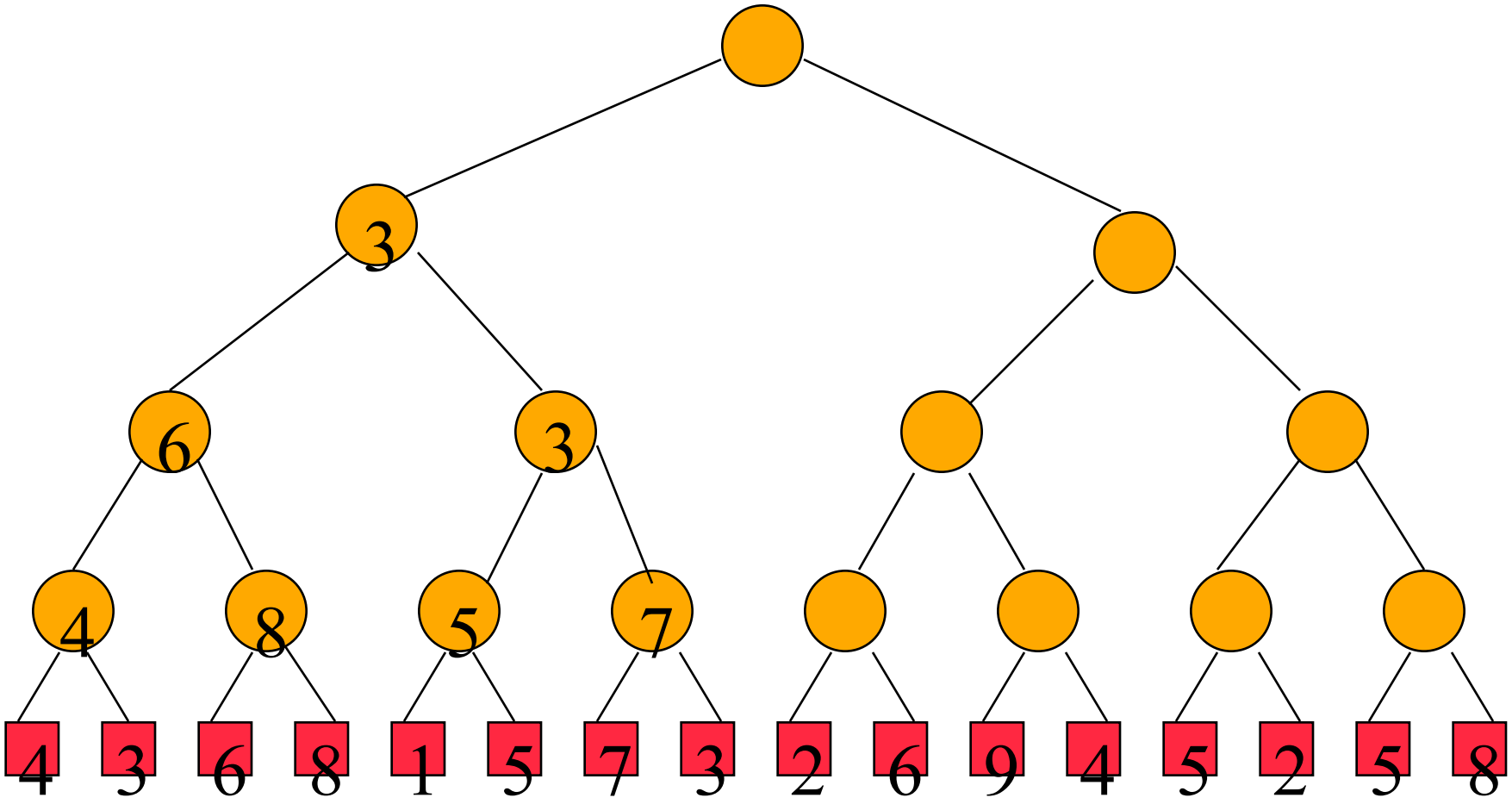

1
1
2
3
1
2
2
3
6
1
3
2
4
2
5
4
3
6
8
6
5
7
3
2
6
9
4
5
2
5
8

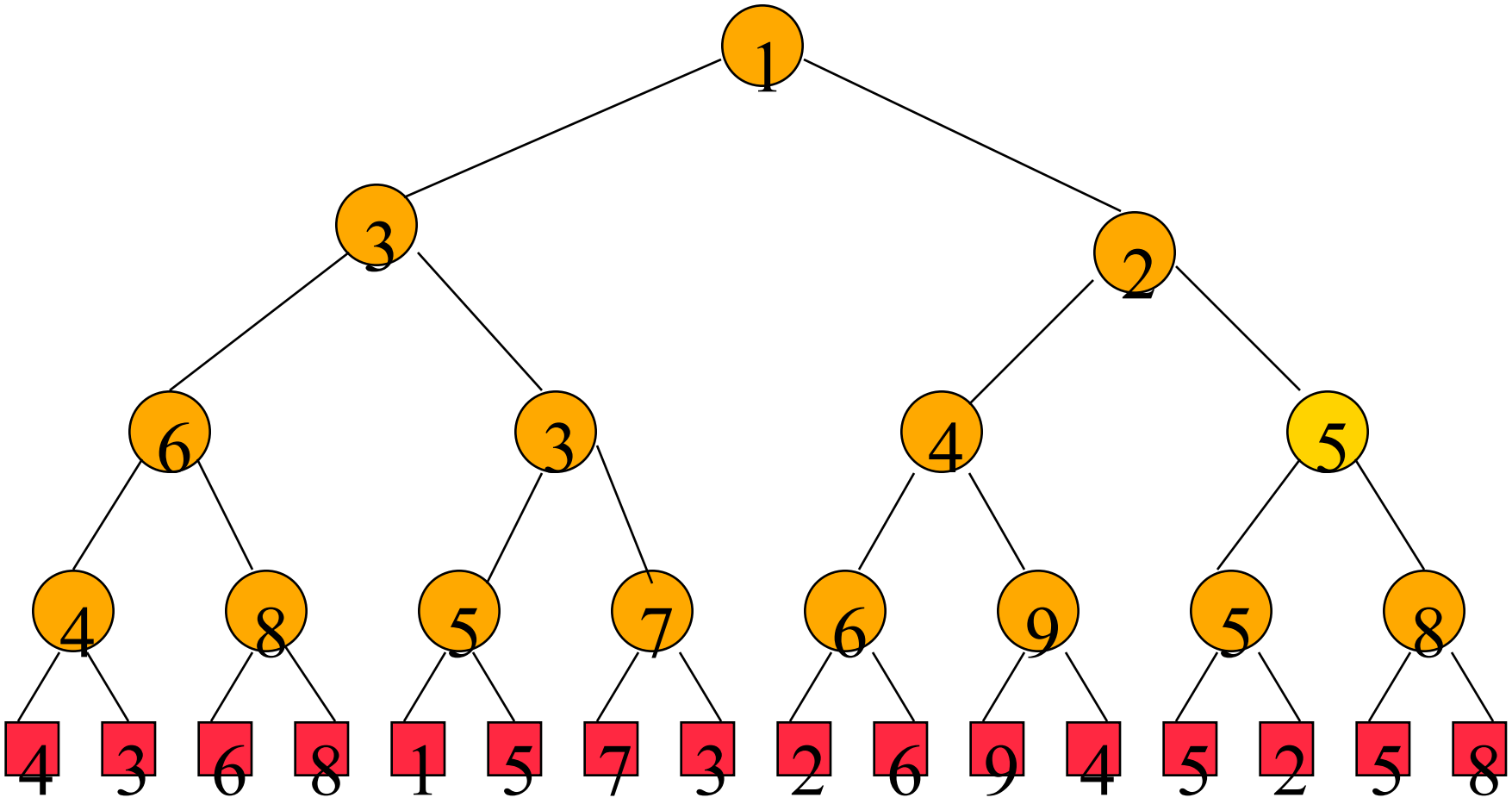


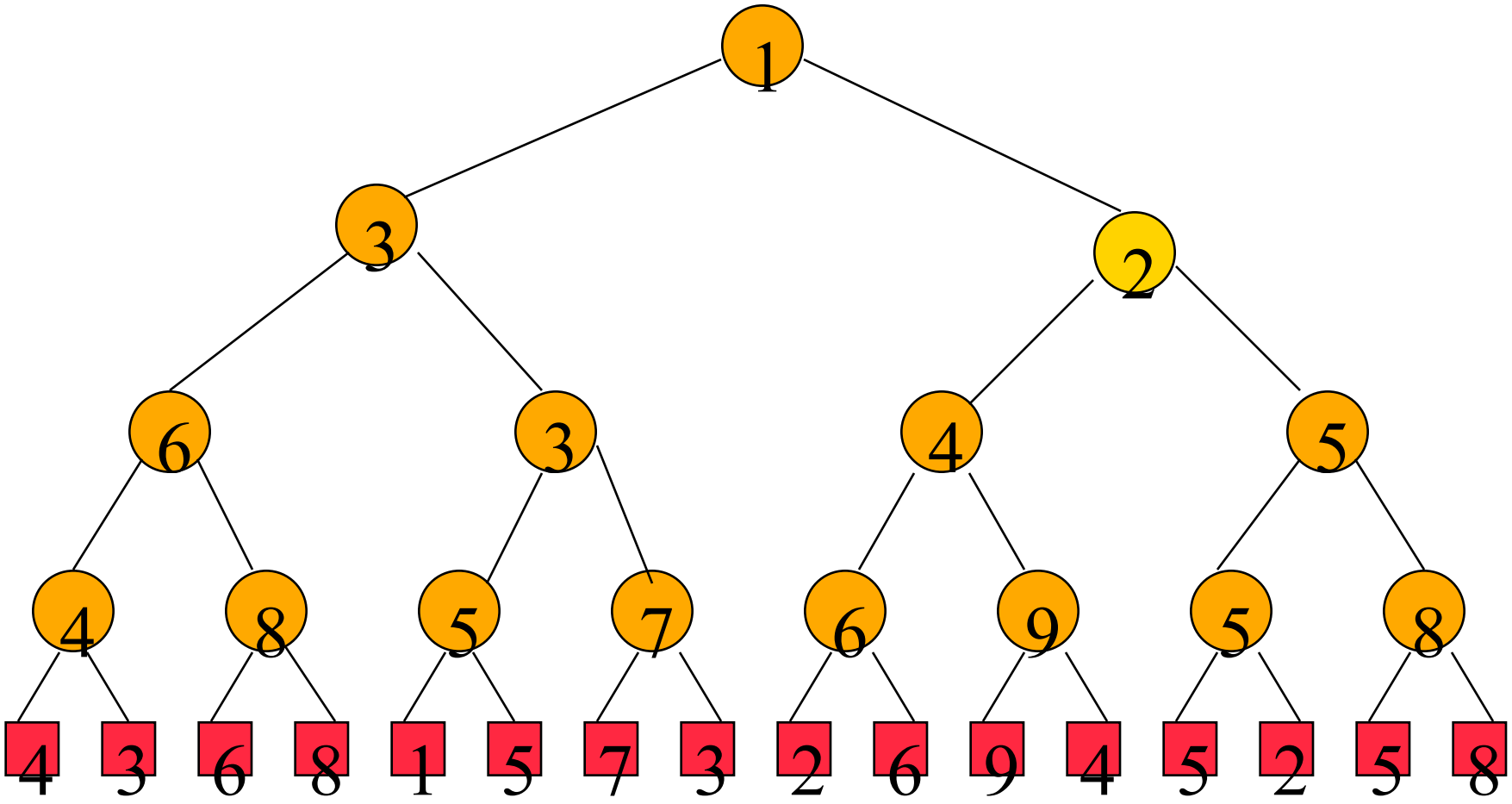

16
3
4
8
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8

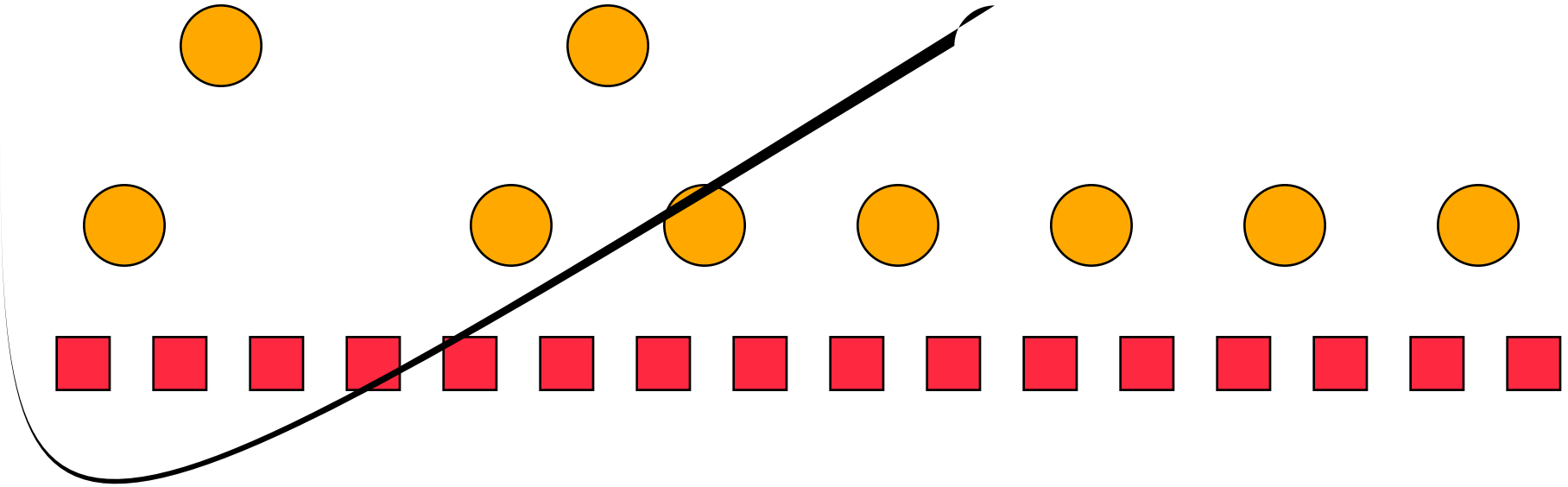

3
6
1
4
8
5
7
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8

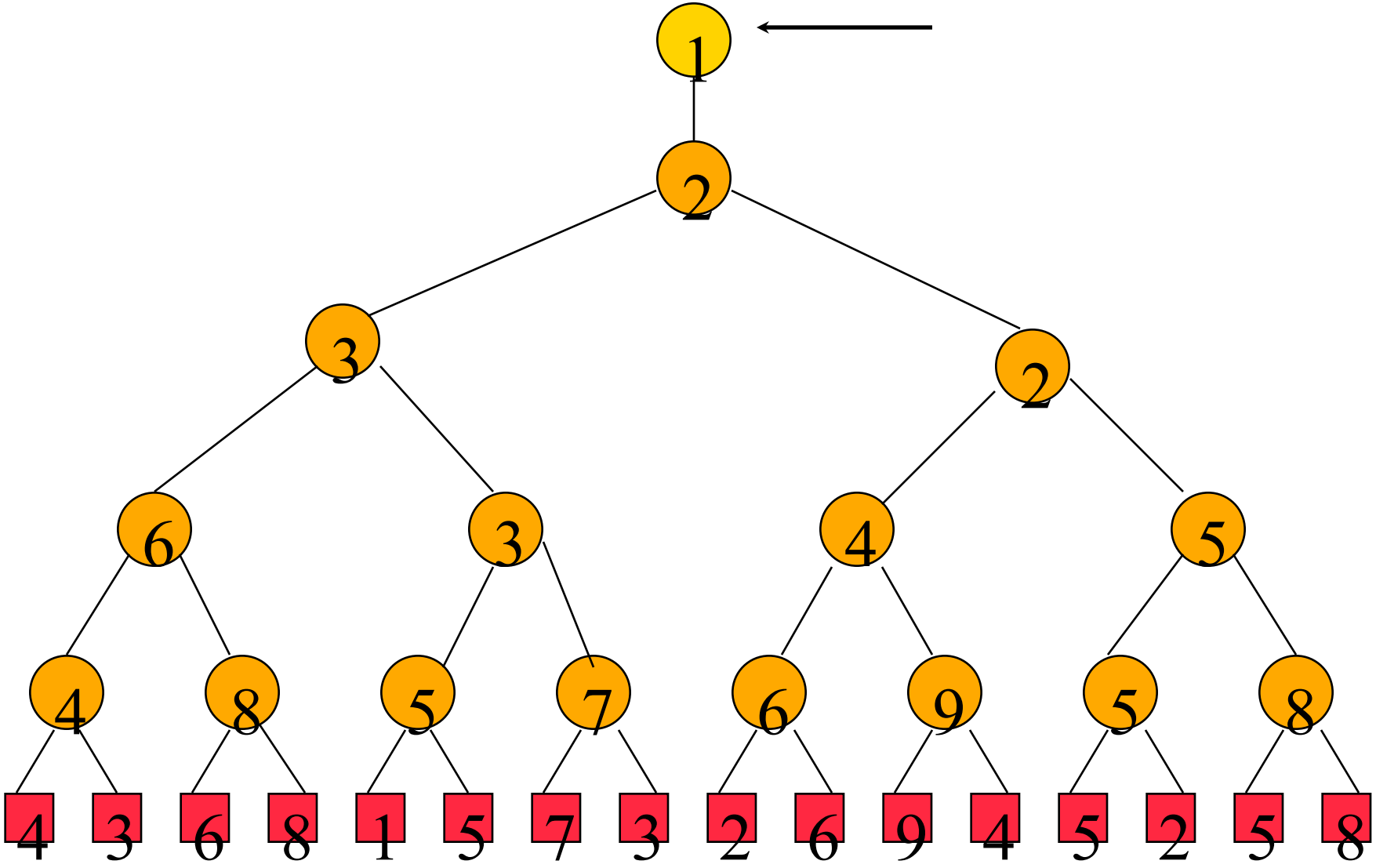

1
3
6
3
2
4
8
5
7
6
9
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8


3
6
3
4
8
5
7
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8


1
3
2
6
3
4
5
4
8
5
7
6
9
5
8
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8


1
3
2
6
3
4
5
4
8
5
7
6
9
5
8
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8




1
2
3
2
6
3
4
5
4
8
5
7
6
9
5
8
4 3 6 8 1 5 7 3 2 6 9 4 5 2 5 8

•

•

( ).

( ).

.

( )

( ).

( )

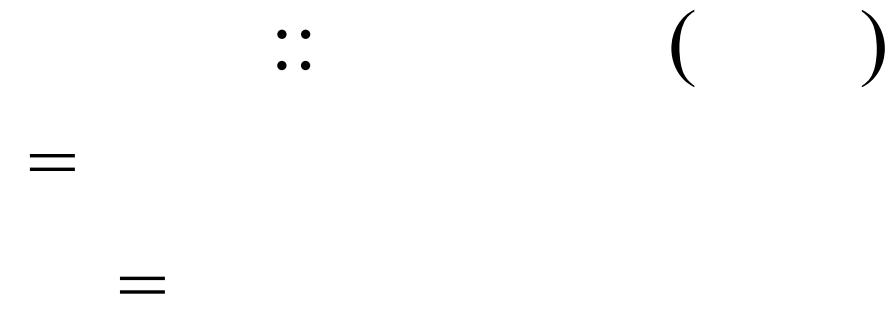
( )

*

*

( )

( )



```
                    ::          ( )


    (  =        1     > 0    --)
    (             (2* ) >              (2*  + 1)
          =                  (2*  + 1)
               =                  (2* )
 0  =   1
    (  = 1    <       ++)
    (      ==                 (2* )       =                  (2*  + 1)
               =                  (2* )
```

:: ( )

(<) .

- .

:: ( )

(<) ( )

-




-
-
-
-

$= 5$

$2, 5, 6, 3, 4$

$= 10$

$= 5$

$2, 5, 6, 3, 4$

$= 10$

$1 = 2, 5$

$2 = 6, 3$

$3 = 4$

$3 \qquad\qquad 2$

$= 5$

$2, 5, 6, 3, 4$

$= 10$

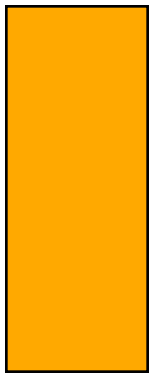
$1 = 2, 5, 3$

$2 = 6, 4$

$= 4$

$= 4, 7, 3, 6$

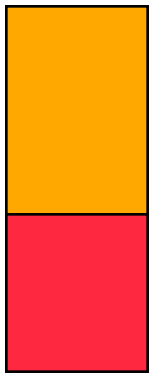
$= 10$



$= 4$

$= 4, 7, 3, 6$

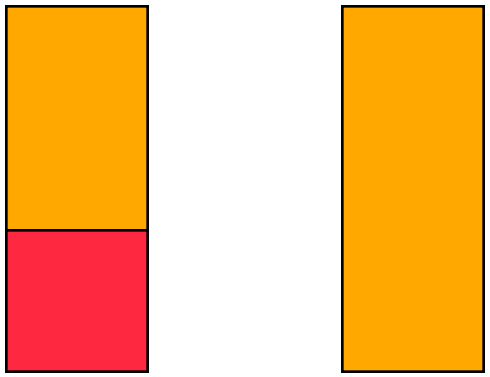
$= 10$



$= 4$

$= 4, 7, 3, 6$

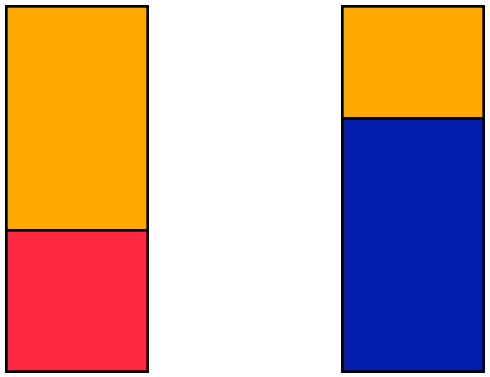
$= 10$



$= 4$

$= 4, 7, 3, 6$

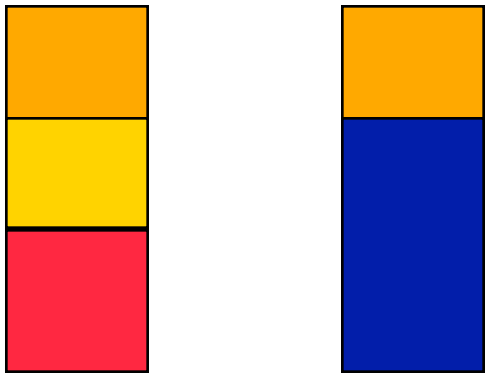
$= 10$



$= 4$

$= 4, 7, 3, 6$

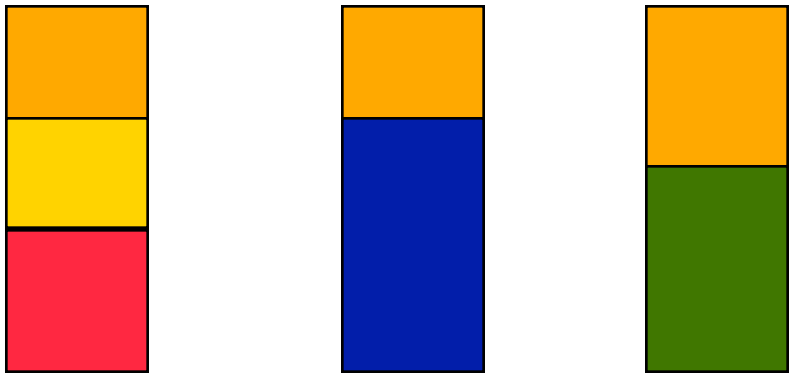
$= 10$



$= 4$

$= 4, 7, 3, 6$

$= 10$

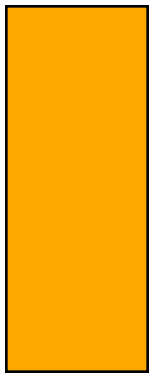


2

$= 4$

$= [4, 7, 3, 6]$

$= 10$

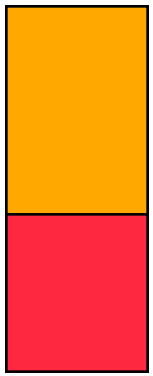


Pack red item into first bin.

$= 4$

$= [4, 7, 3, 6]$

$= 10$



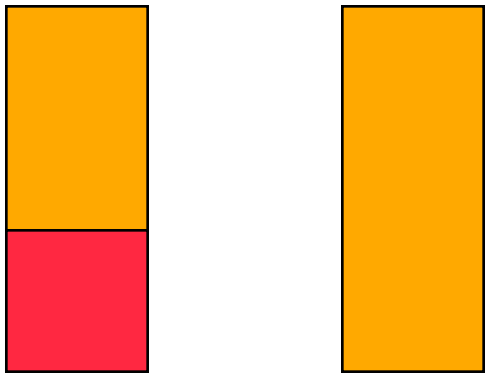
Pack blue item next.

Doesn't fit, so start a new bin.

$= 4$

$= [4, 7, 3, 6]$

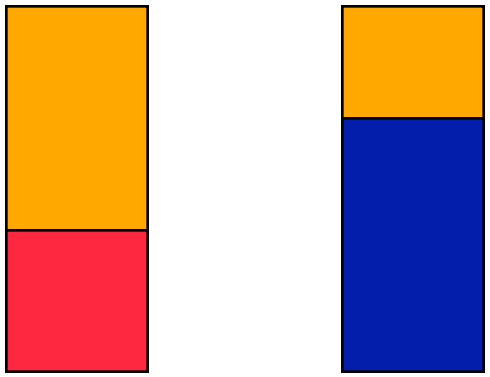
$= 10$



$= 4$

$= [4, 7, 3, 6]$

$= 10$

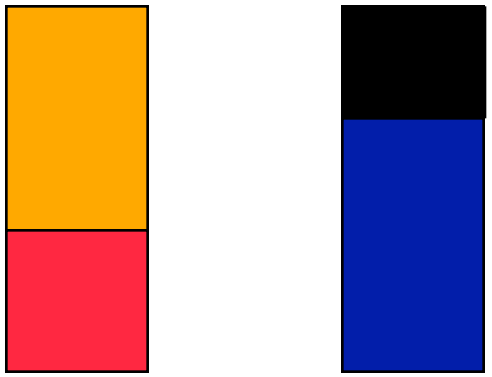


Pack yellow item into second bin.

$= 4$

$= [4, 7, 3, 6]$

$= 10$

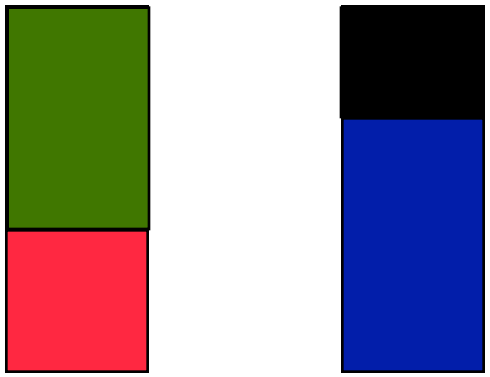


Pack green item into first bin.

$= 4$

$= [4, 7, 3, 6]$

$= 10$



Optimal packing.

(,).

.

>= .

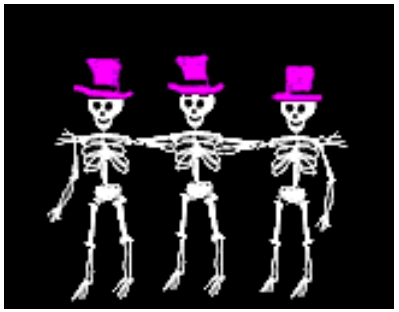
.

.

,

.

.



:

<= (17/10)( ) + 2

:

<= (11/9)( ) + 4

.

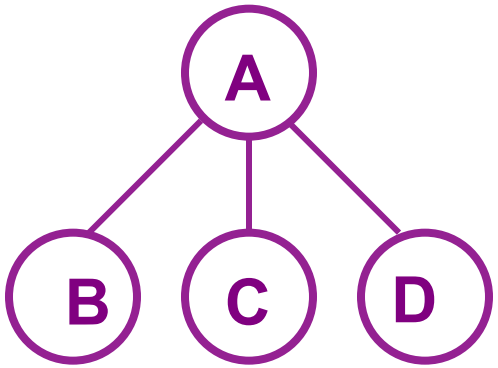
( ),

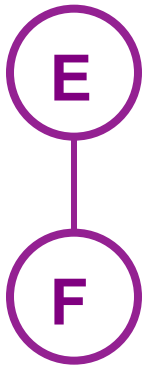
.

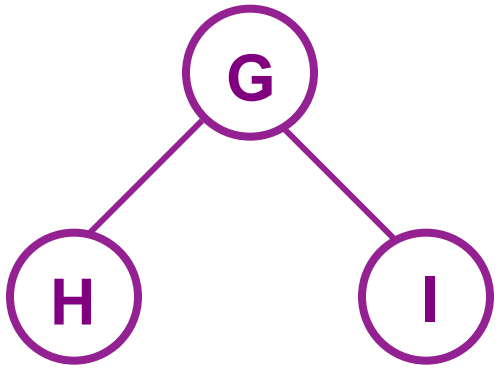
- **Exercises: P301-1,4**

:

- ≥ 0







:

1, , ,

, (1, ,),

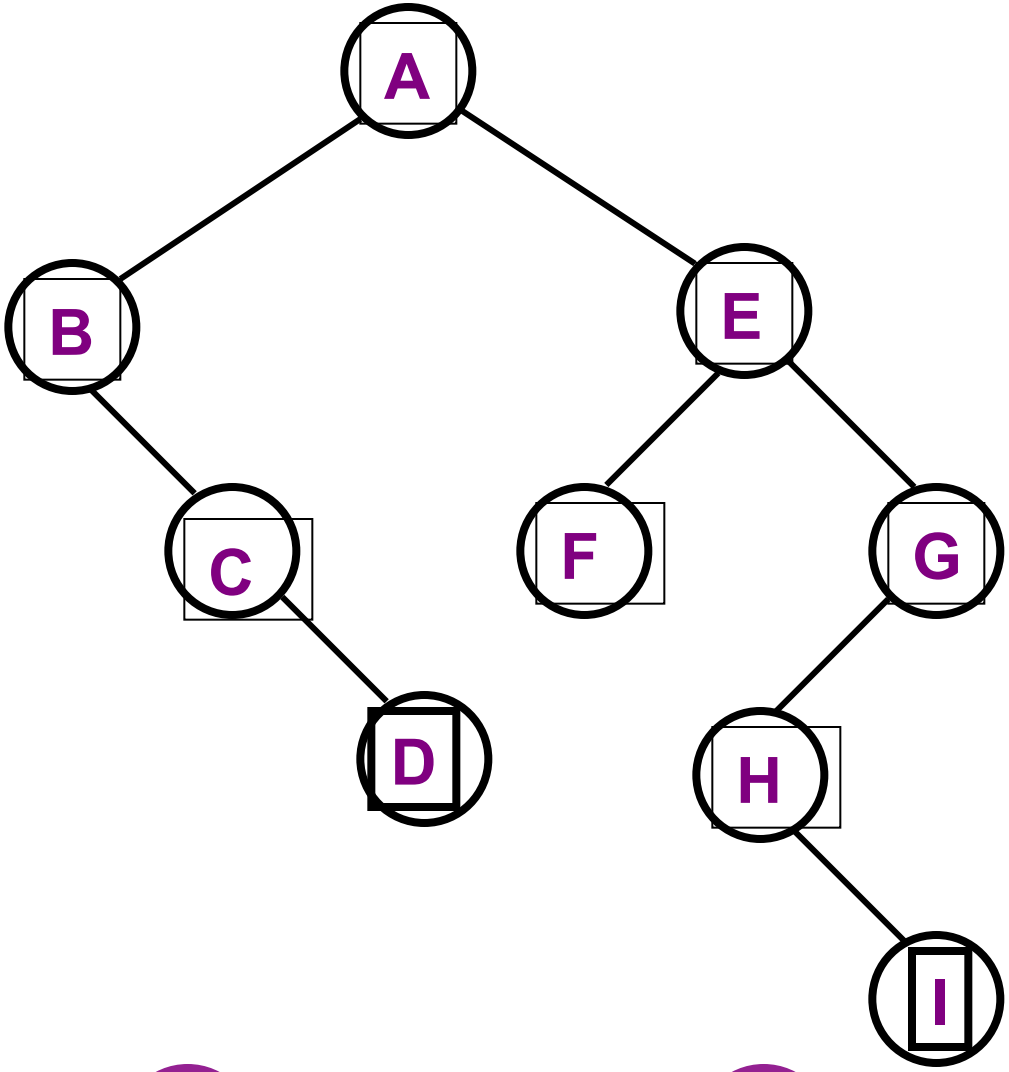
(1) =0

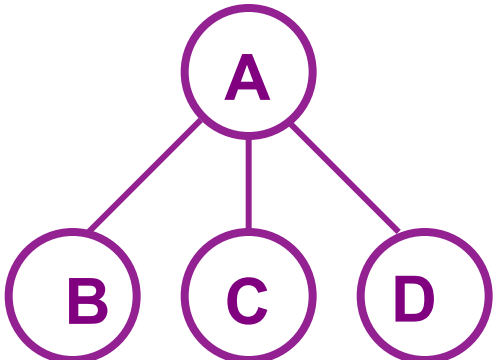
(2) (1);

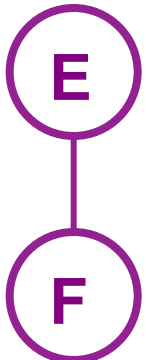
(11, , 1), 11, , 1

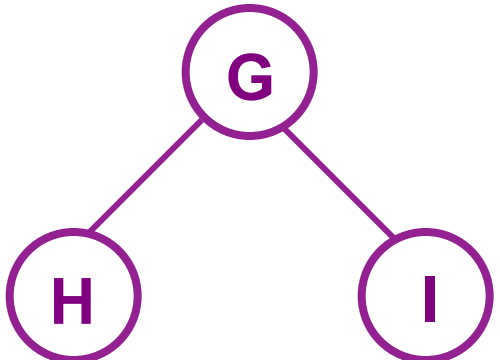
(1); (2, ,

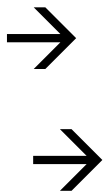
).

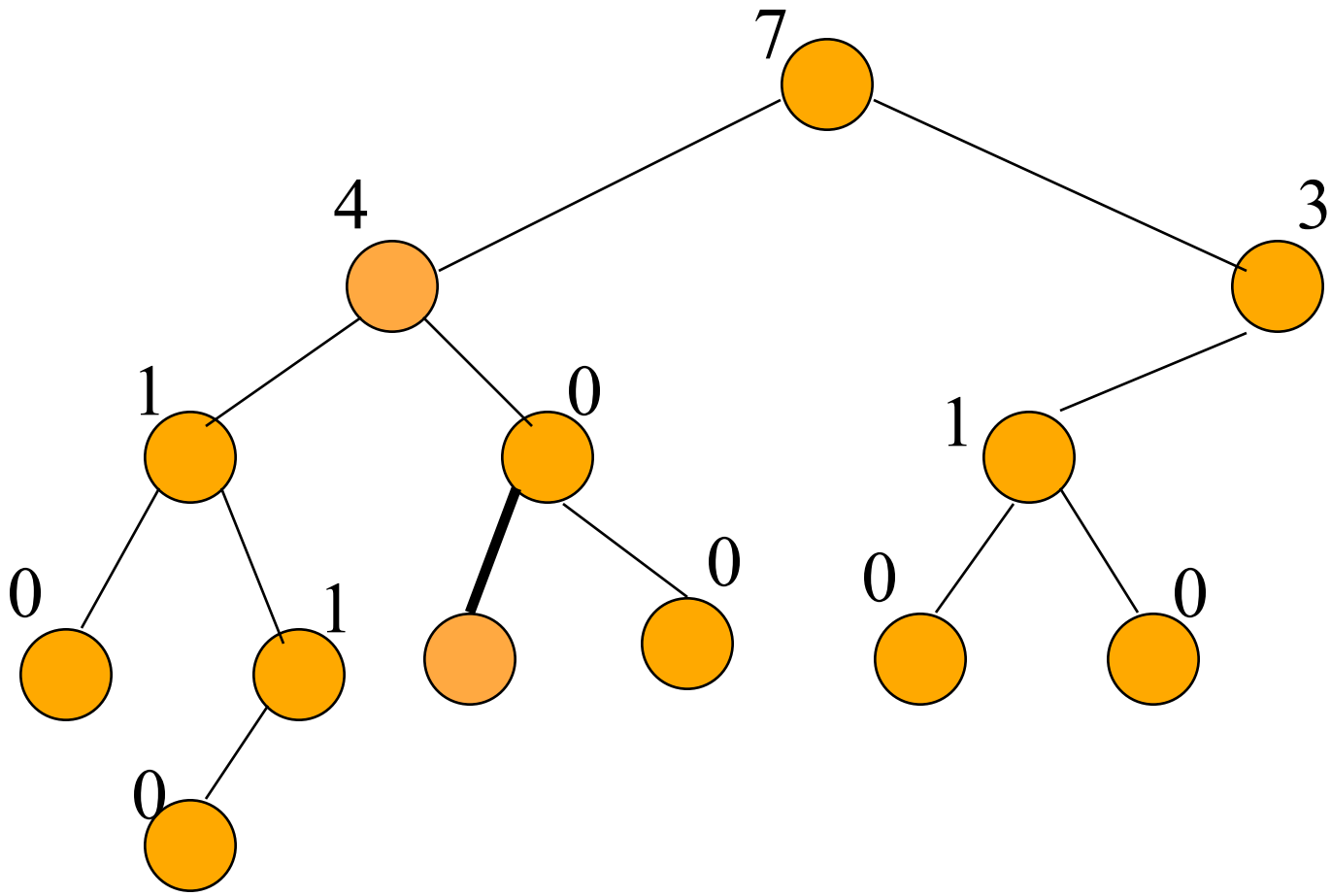

A
B
E
C
F
G
D
H
I


A
B
C
D


E
F


G
H
I




7
4
3
1
0
1
0
0
0
1
0
0

.

:

(1) .

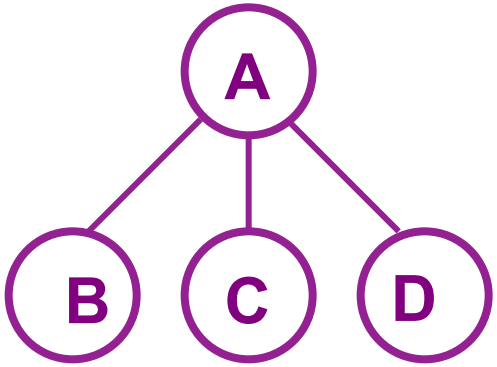
(2) .

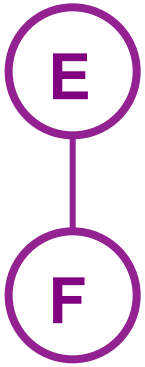
(3)

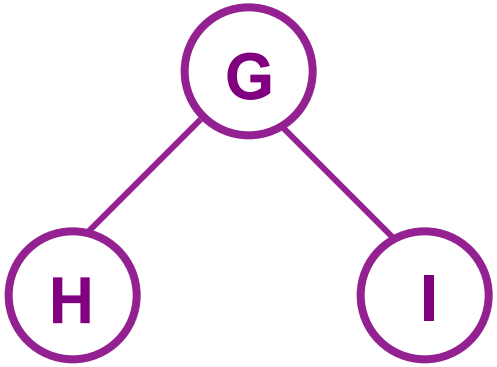
.

(4)

.


A
B
C
D


E
F


G
H
I

:

(1) .

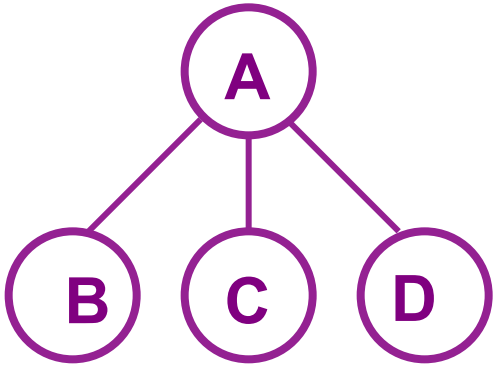
(2)

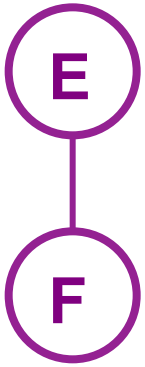
.

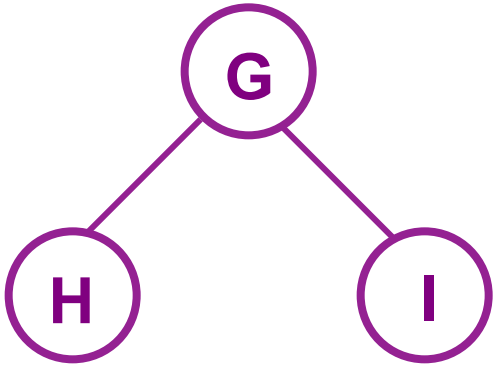
(3) .

(4)

.


A
B
C
D


E
F


G
H
I

:

(1) .

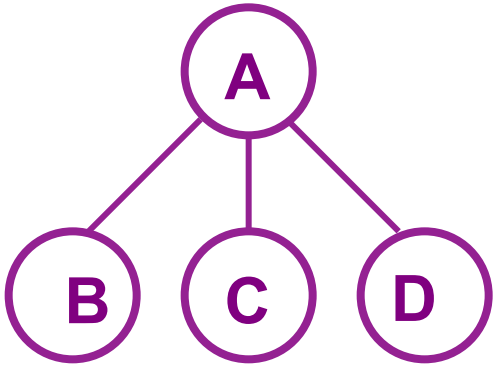
(2)

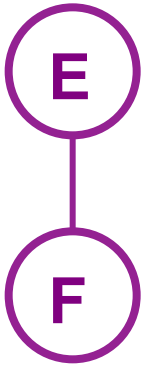
.

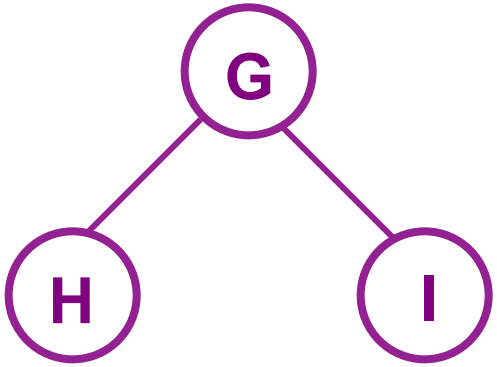
(3)

.

(4) .

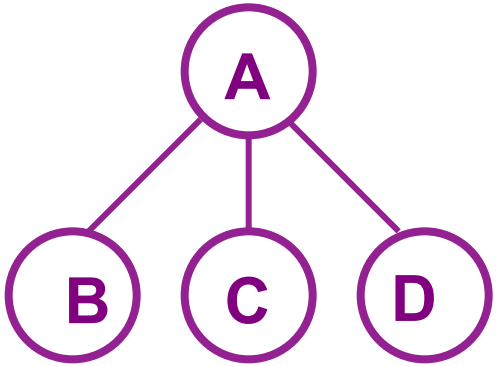

A
B
C
D

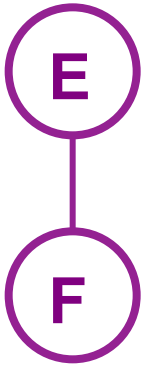

E
F

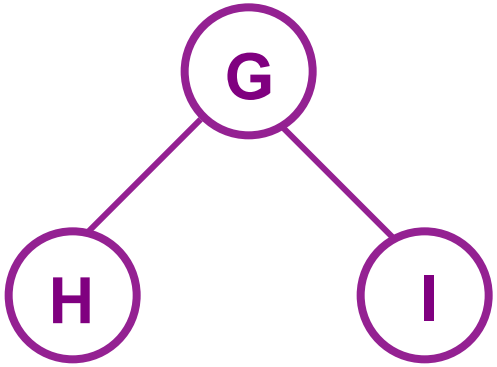

G
H
I

-

, .


A
B
C
D


E
F


G
H
I

: 304-3.

:

0, 1, 2,

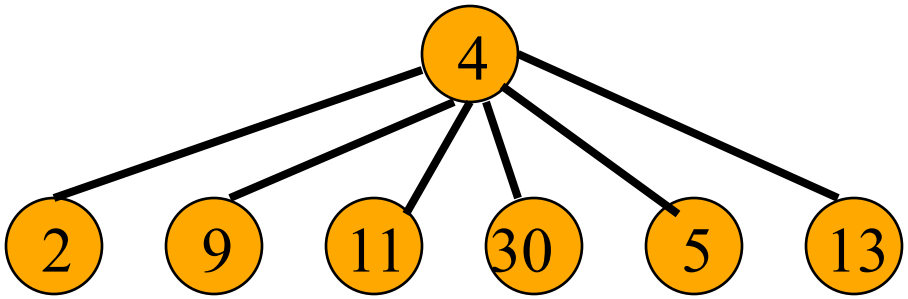
, -1 ( ).

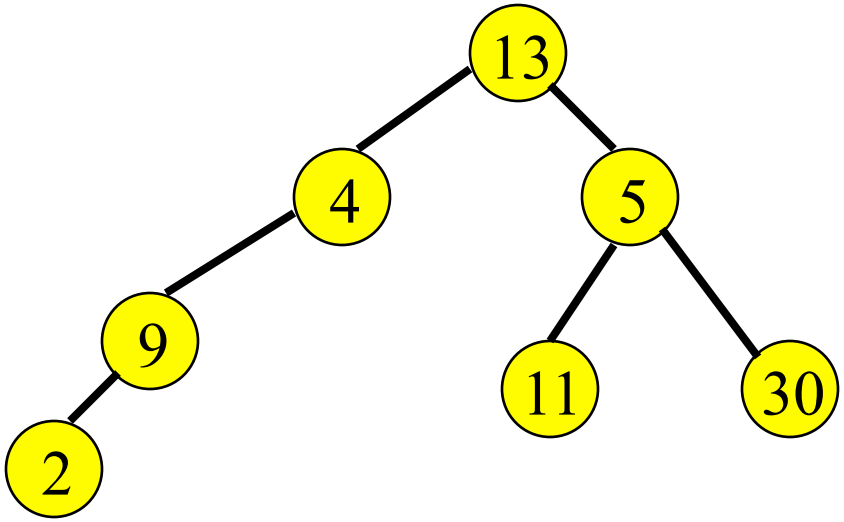
, , ≠ , ∩ =∅.

:

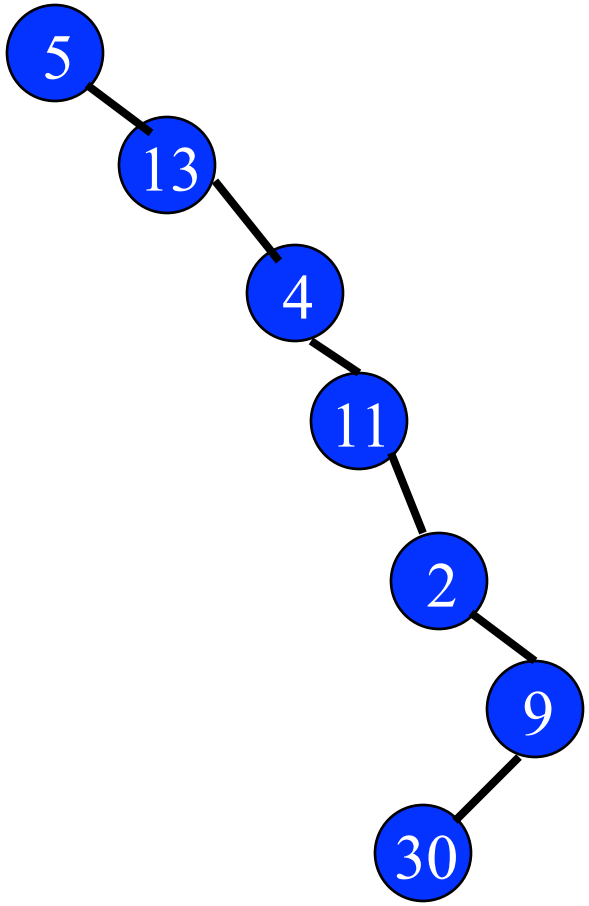
- ∪ .
- ( )--- .

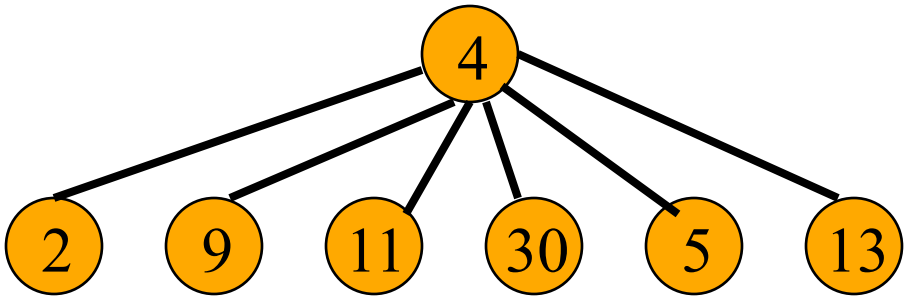
$= 2, 4, 5, 9, 11, 13, 30$



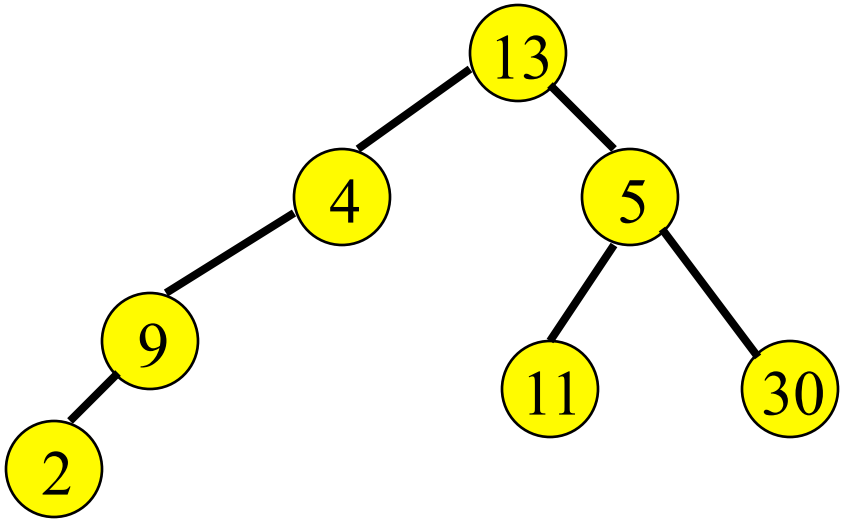


:

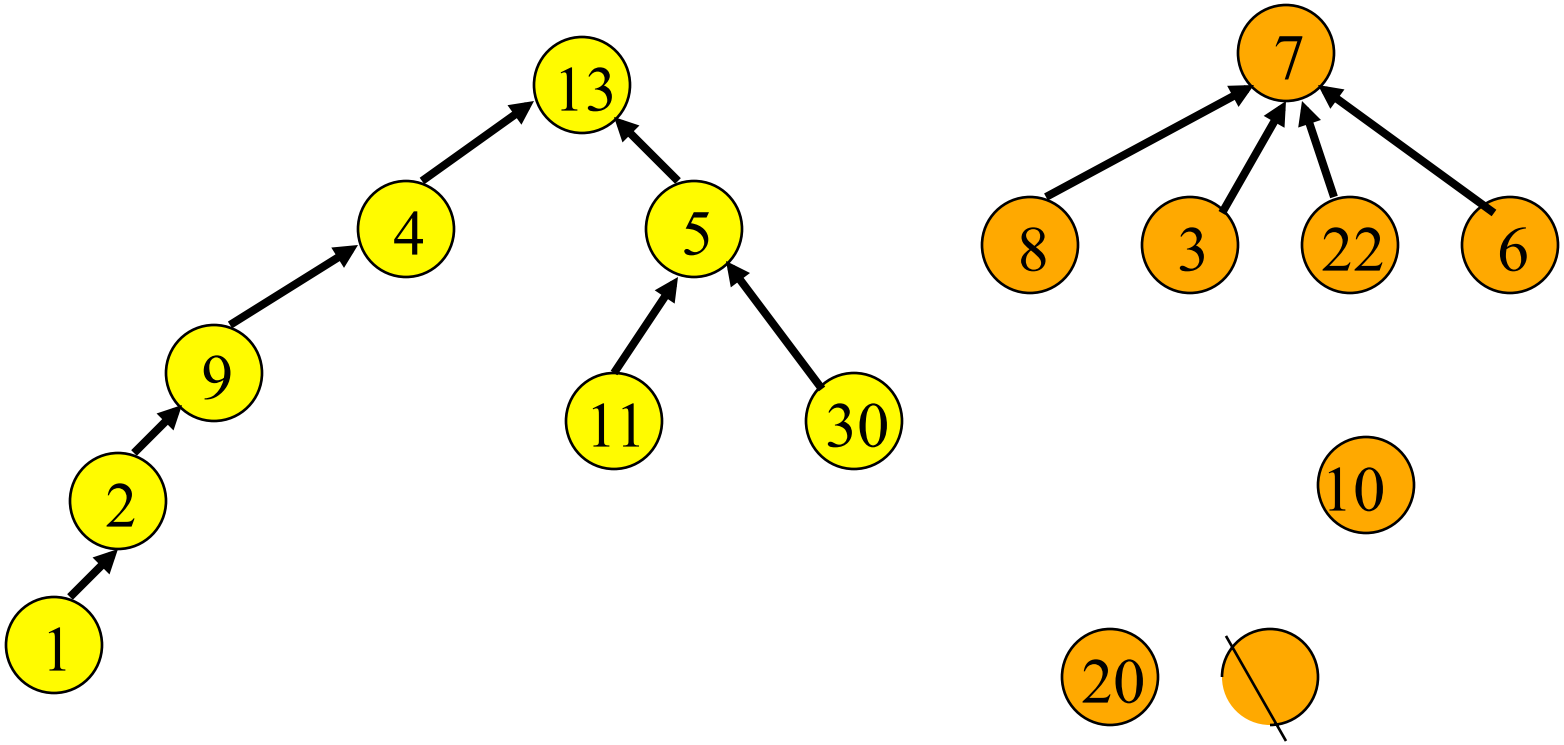


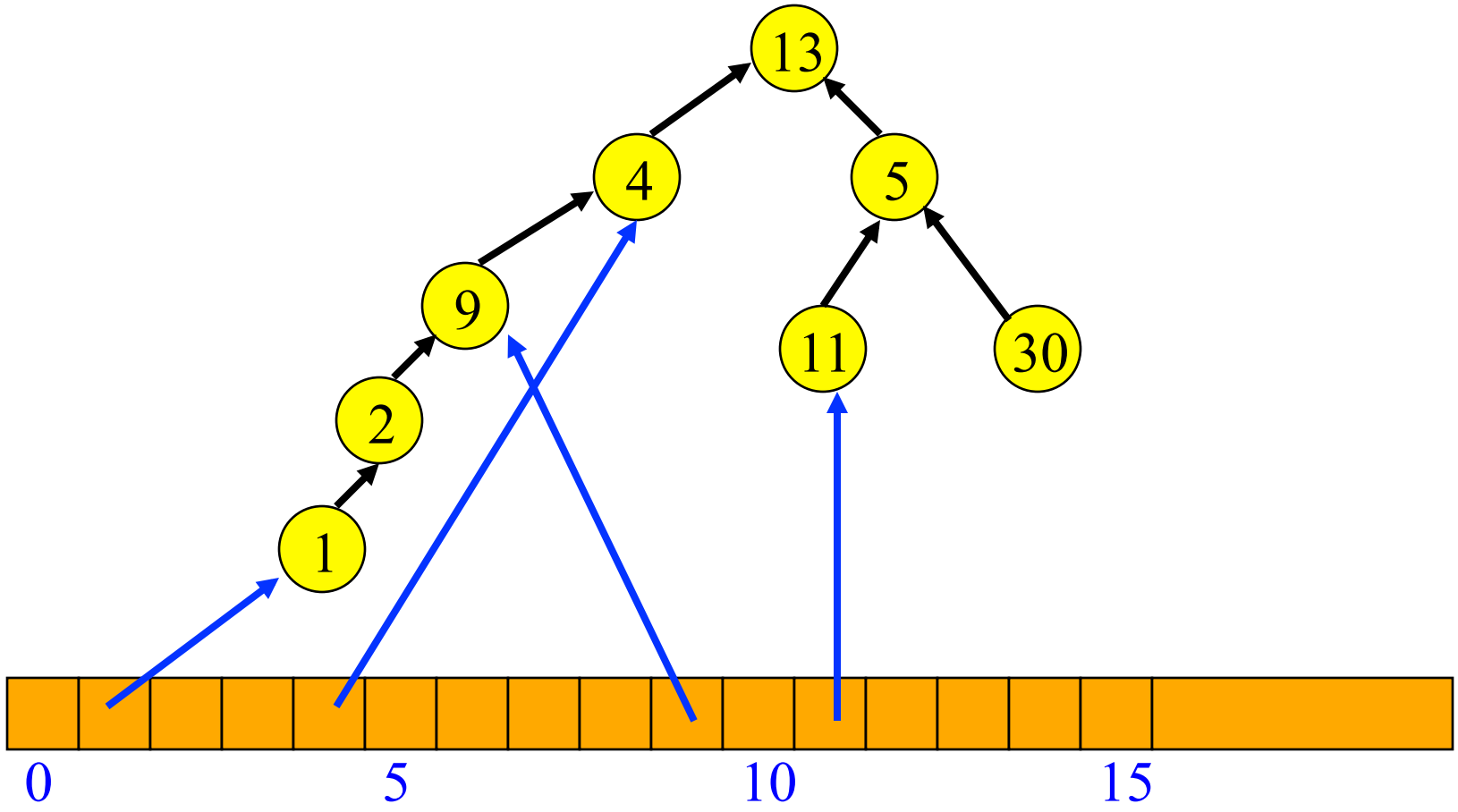

4
2
9
11
30
5
13

( )

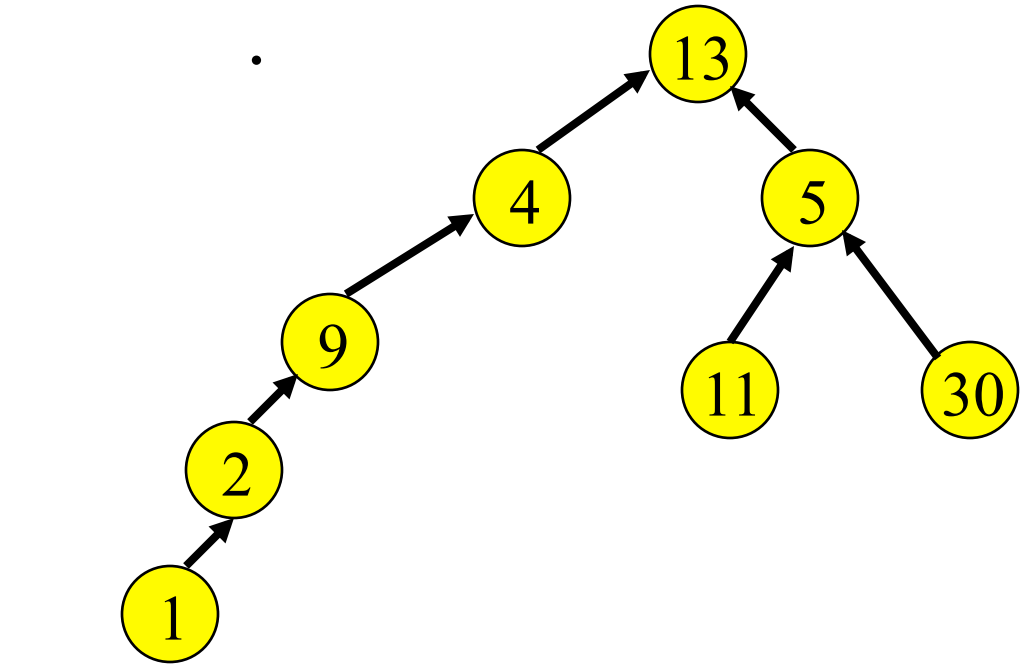

13
4
5
9
11
30
2

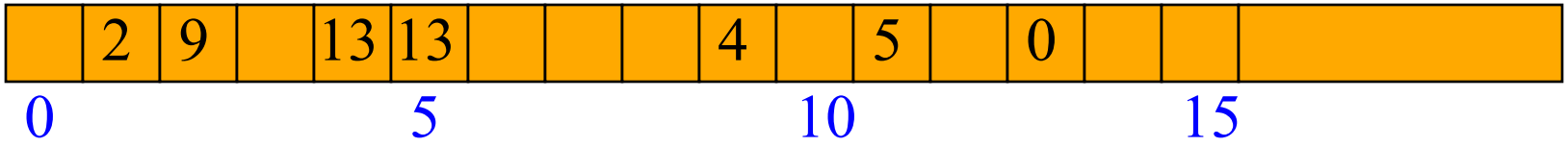
( )


13
4
5
9
11
30
2
1
7
8
3
22
6
10
20


13
4
5
9
11
30
2
1
0
5
10
15

(.)


13
4
5
9
11
30
2
1


2
9
13
13
4
5
0
0
5
10
15

//

*

//

:: ( )

(< 2) 2

.

=

=

(, + , -1)

(,)

- , != .

,

.

- =

:: ( )

// .

(>=0) =

::

(,)

//

,

!=

//

=

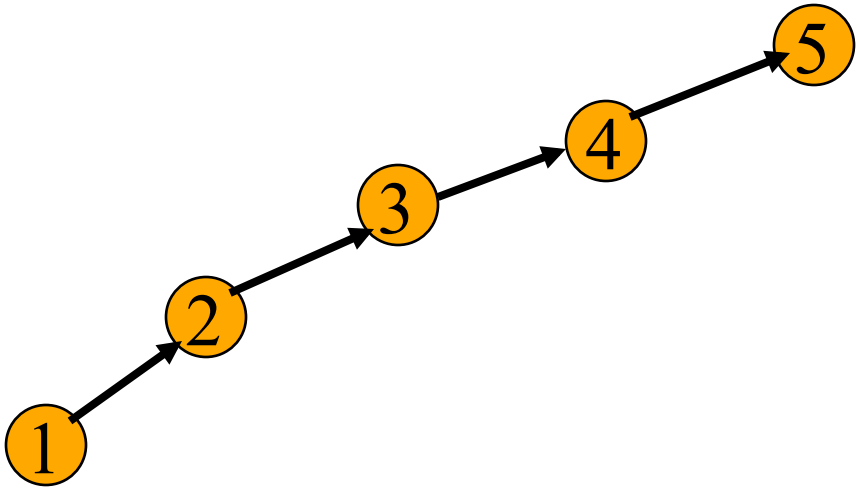
()

(1)

()

.

- (2,1), (3,2), (4,3), (5,4)



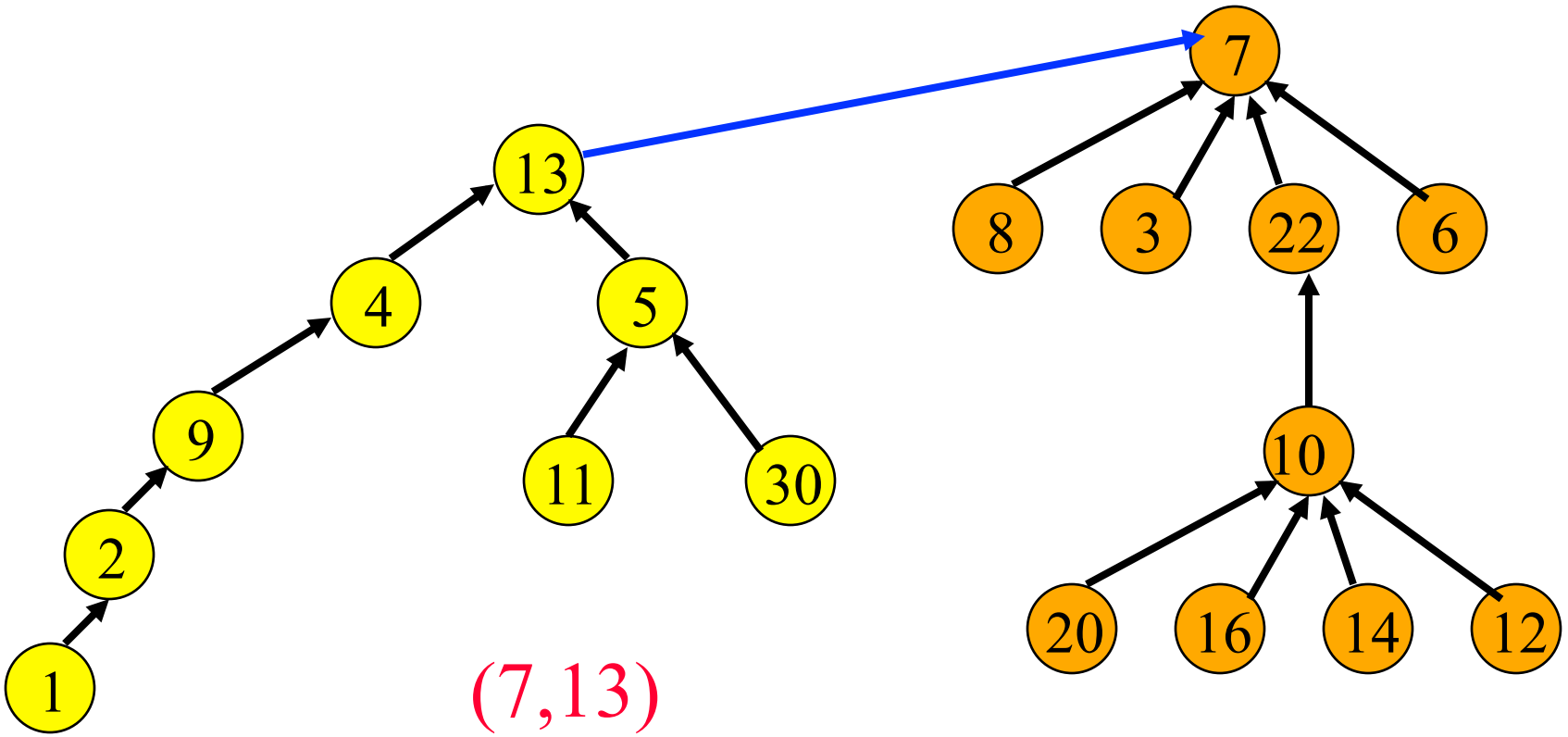
( ).

(+) = ( )

= 0

( ).

(+).


7
13
8
3
22
6
4
5
9
11
30
10
2
20
16
14
12
1
(7,13)

:: (,)

// , ≠ ,

// = - = -

= +

(>) //

=

=

//

=

=

.

Lemma 5.5

$$\left(\quad _2 \quad \right) + 1.$$

- **Lemma 5.5**

$$(\quad _2 \quad) + 1.$$

-
-
-
-

≤

≤ ≤

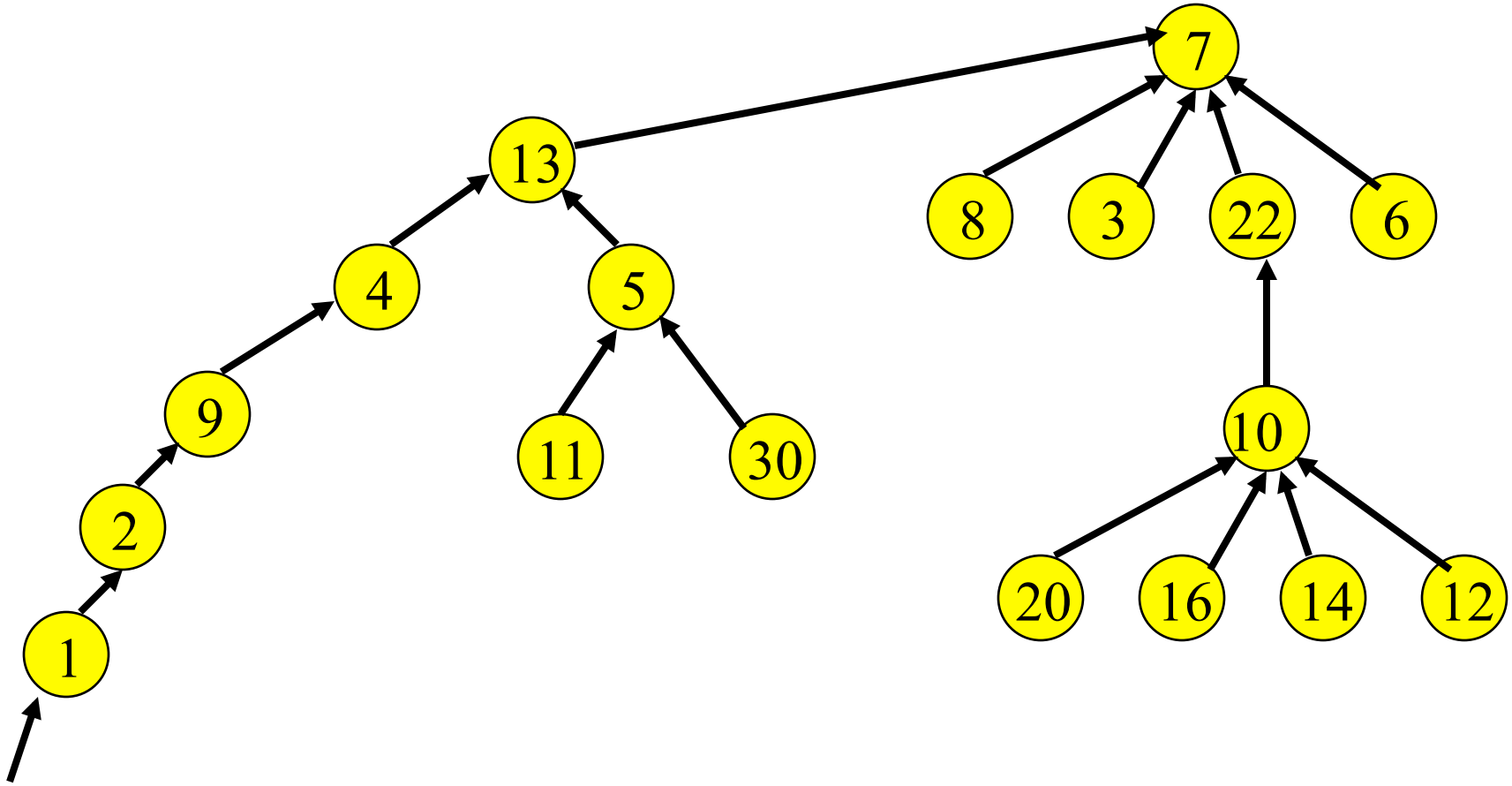
≤ ⌊ ⌋

⌊ ⌋ ≤ ⌊ ⌋

m-a ≥ m/2 ≥ a

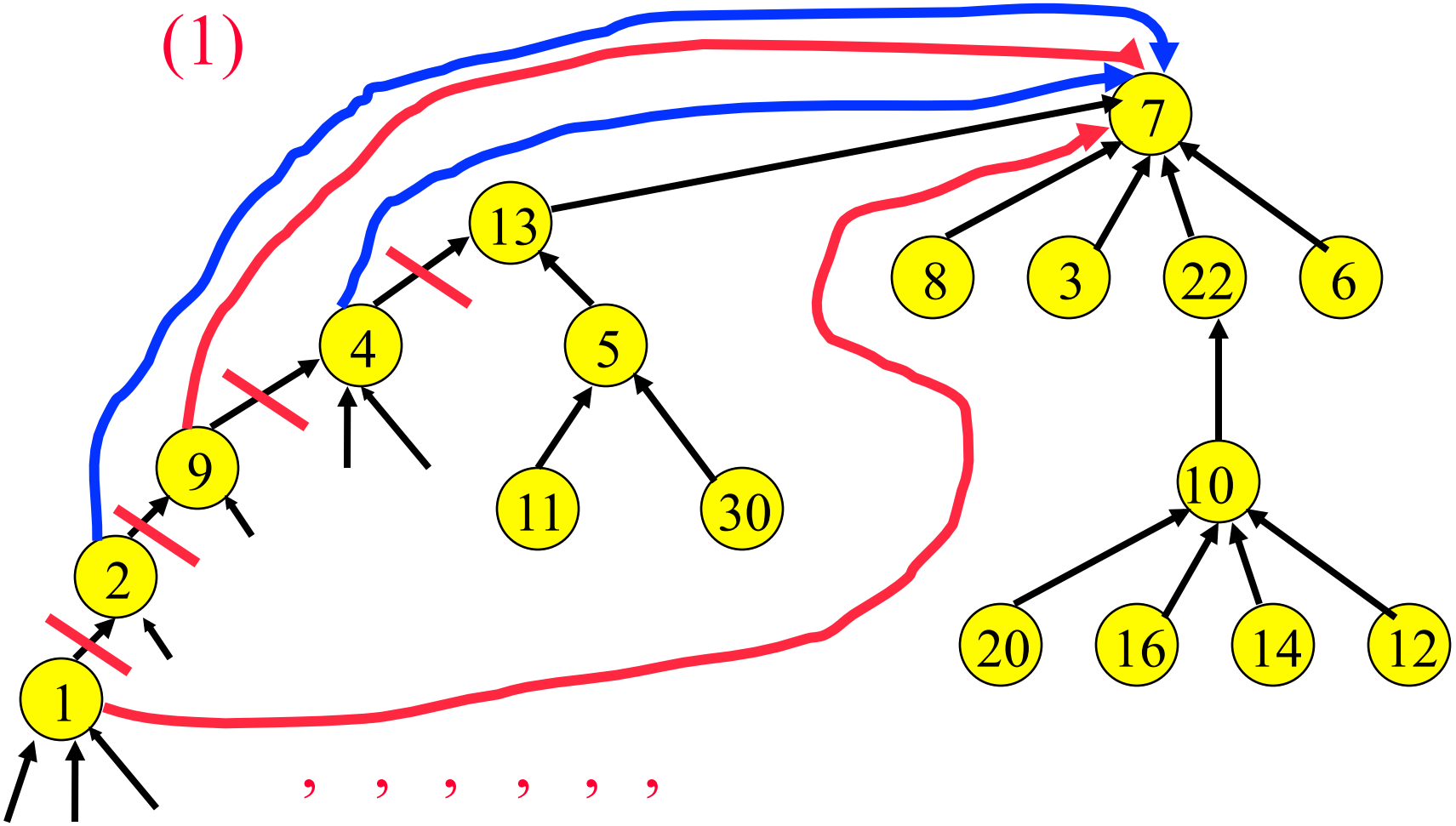
≤ ⌊ ⌋

≤ ⌊ ⌋ ≤



(1)

(1)



:: ( )

// .

// .

//

(= >= 0 =)

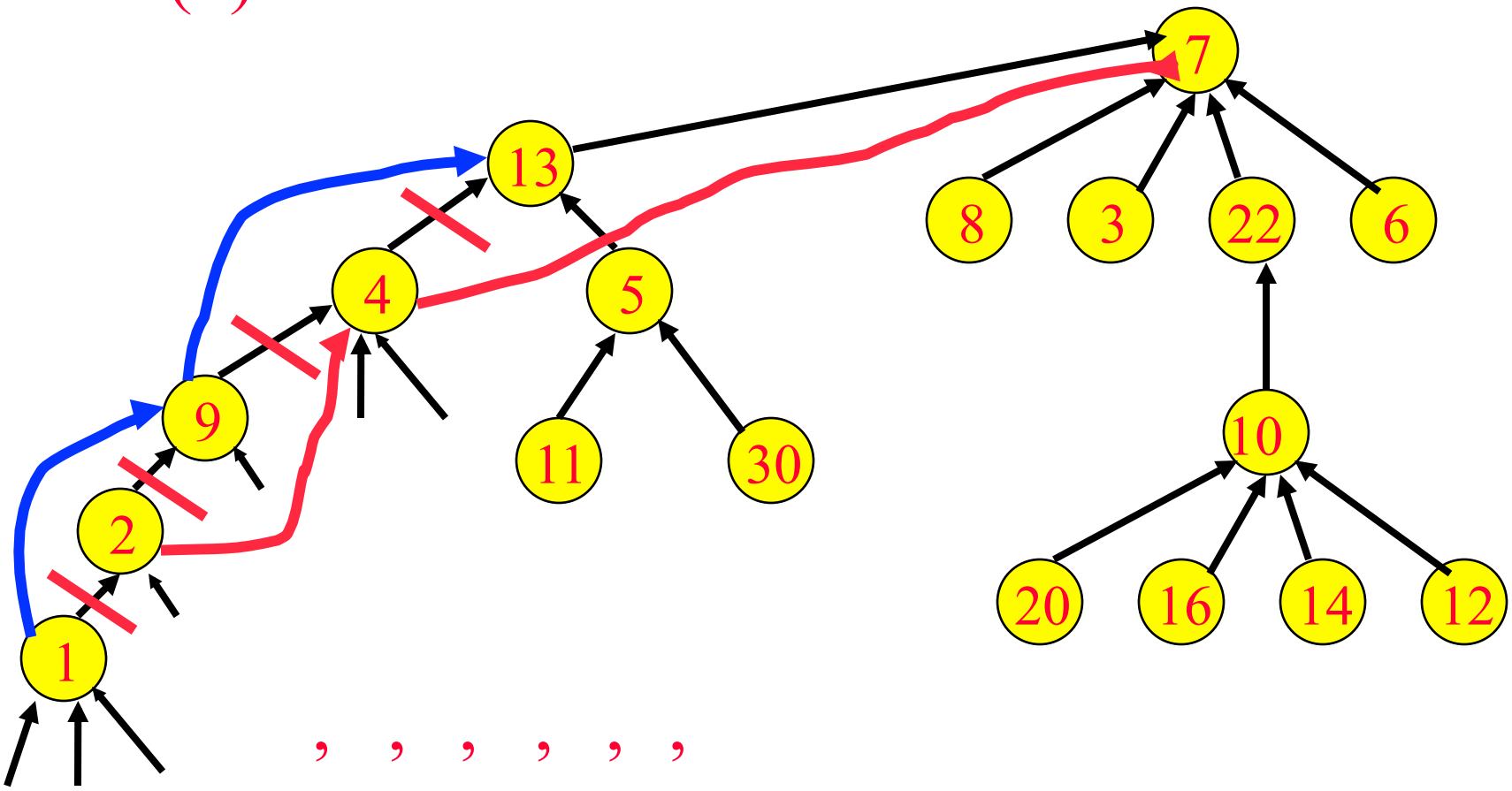
(!=)

=

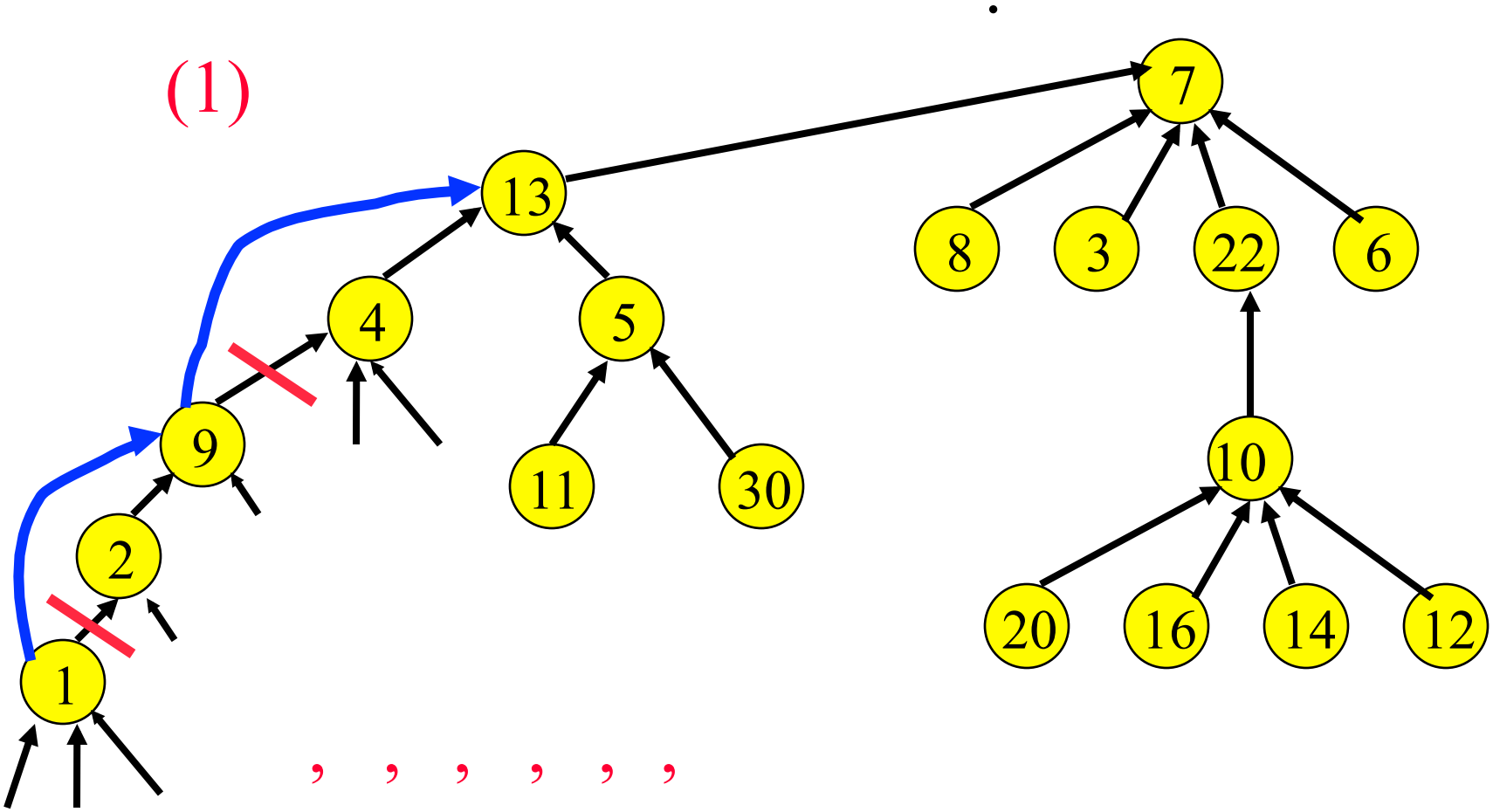
=

=

(1)



, , , , , ,


(1)
7
13
8
3
22
6
4
5
9
11
30
10
2
20
16
14
12
1

$\Leftrightarrow$

$\leq \quad \leq$

$\equiv$

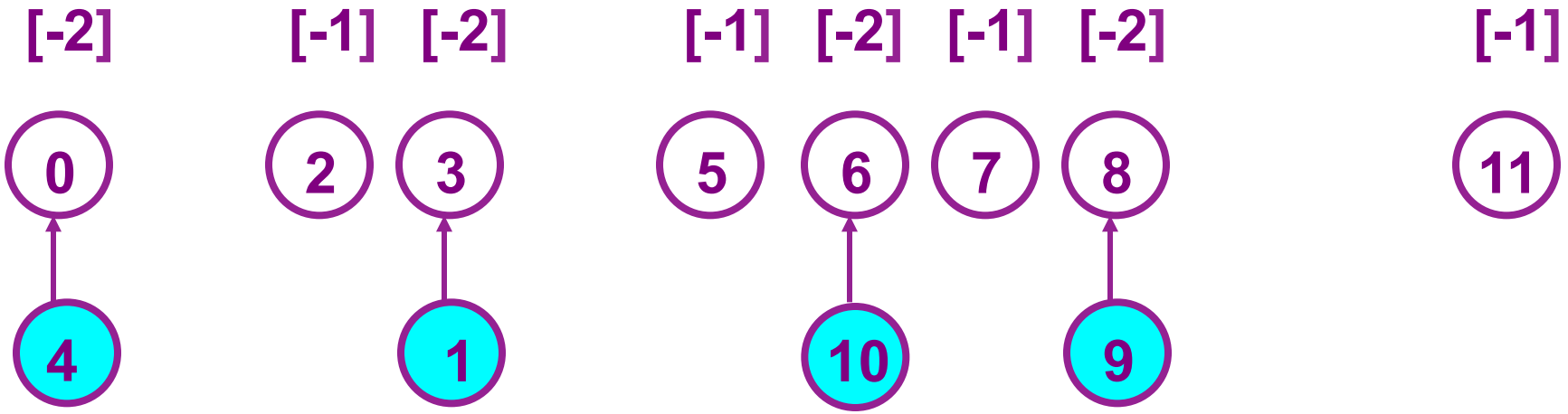
$\neq$

α

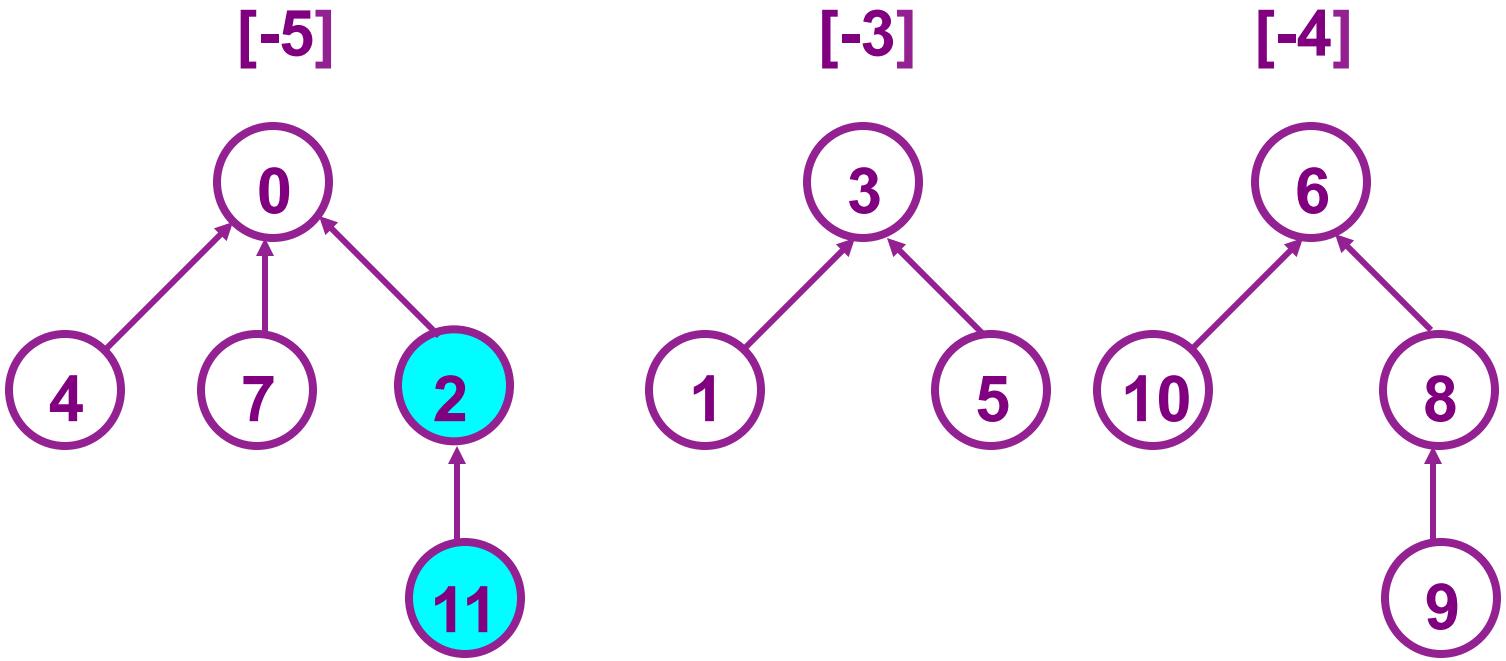
[-1] [-1] [-1] [-1] [-1] [-1] [-1] [-1] [-1] [-1] [-1] [-1]

0 1 2 3 4 5 6 7 8 9 10 11

(a) Initial trees



(b) After processing 0 ≡ 4, 3 ≡ 1, 6 ≡10, and 8 ≡ 9



(d) After processing 11 ≡ 0

- : → 101011100001
- :
- ,

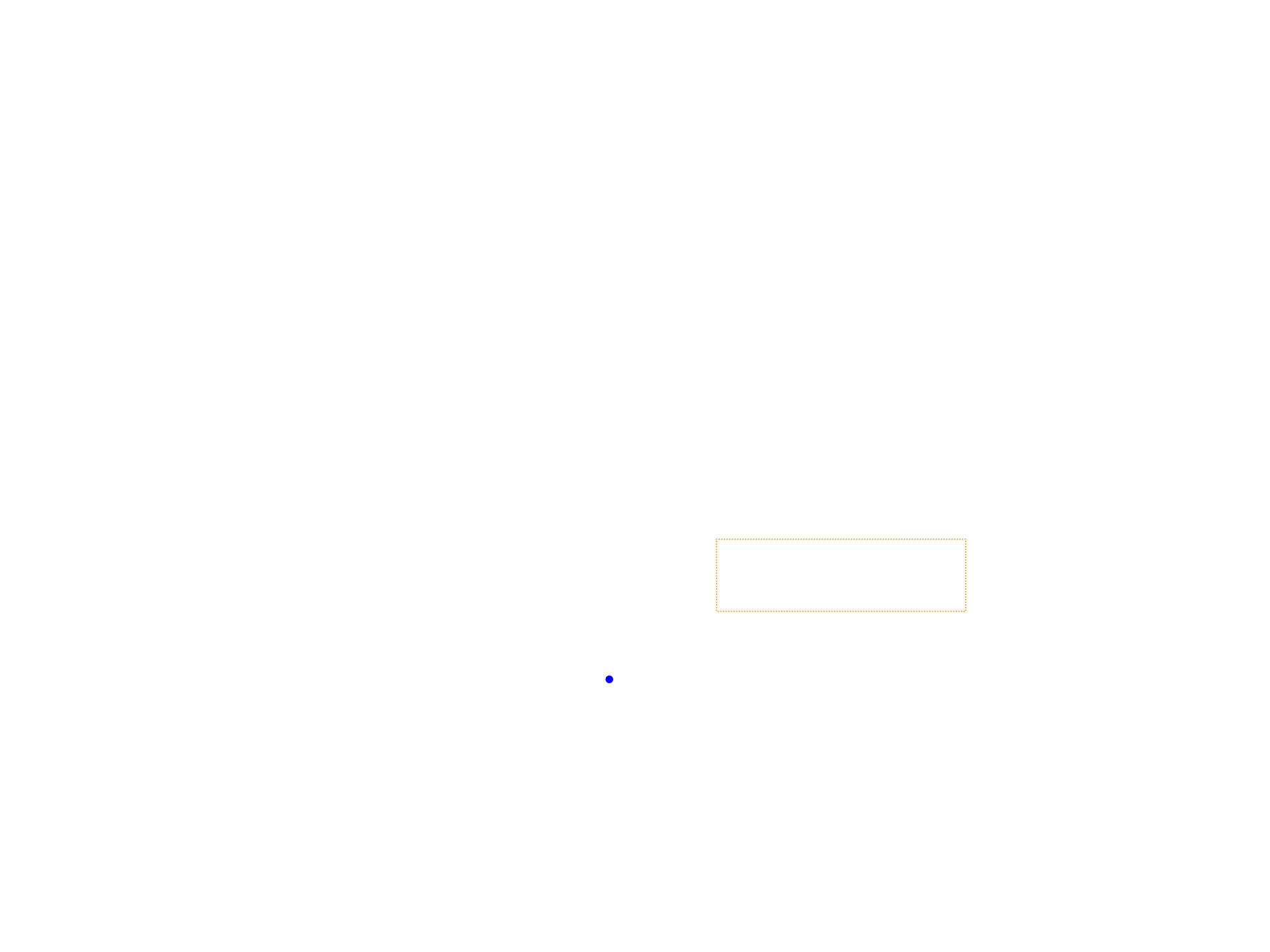
- :
- →
- (Σ( ))

:

-

:

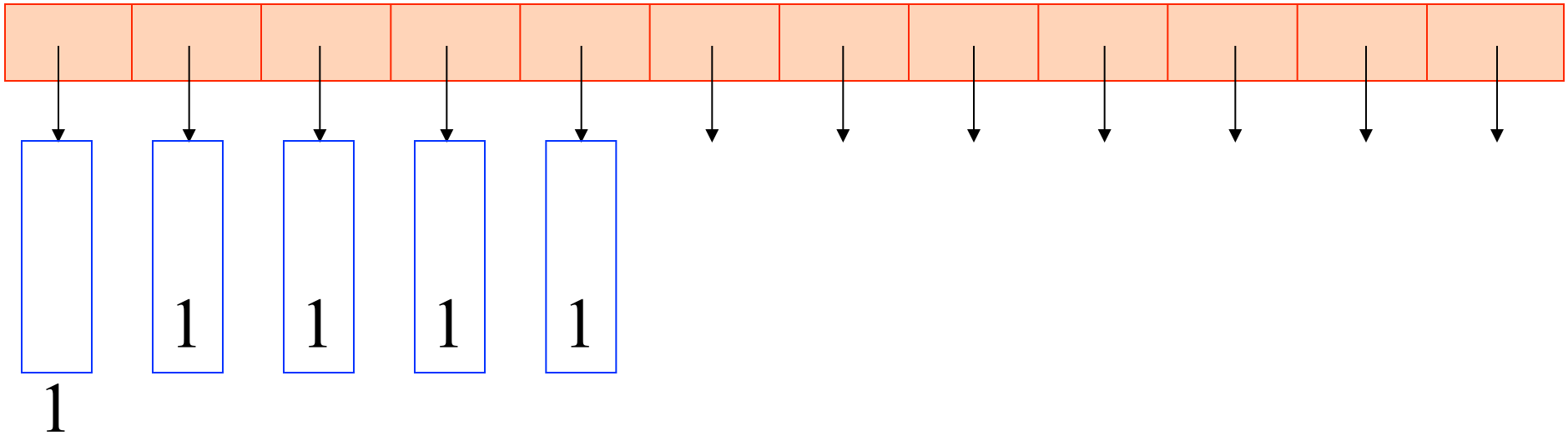
- →
- (Σ( ))

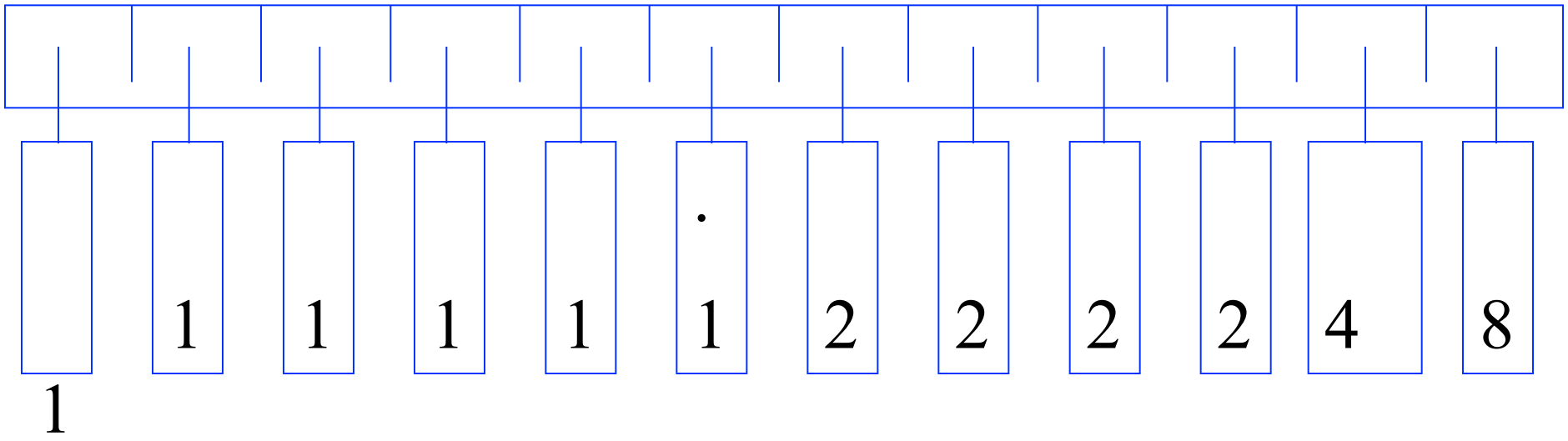


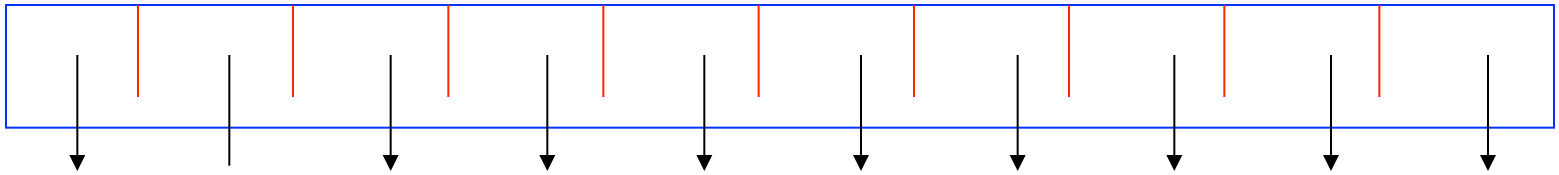
.

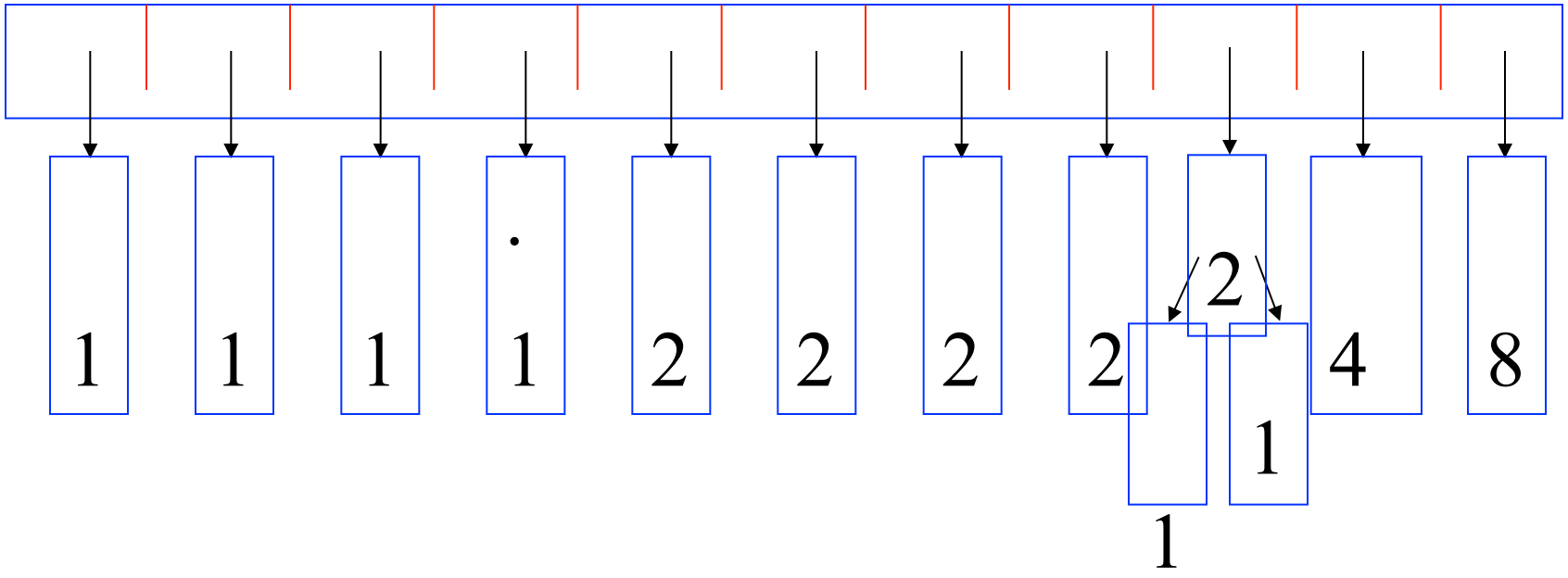
.	.	.
1	1	1
8	2	. 1
2	2	
1	2	
4	1	

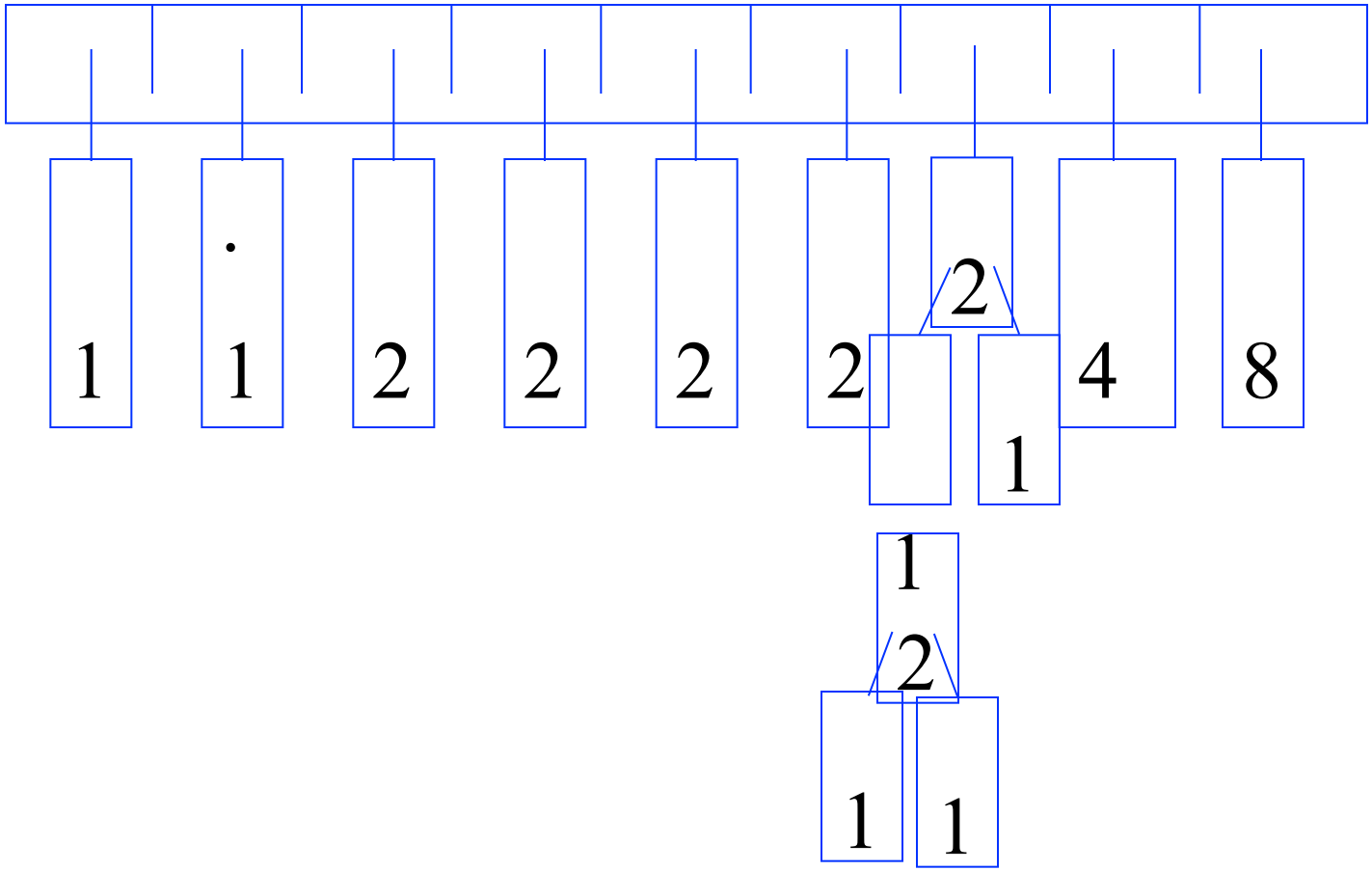
-
-
  -

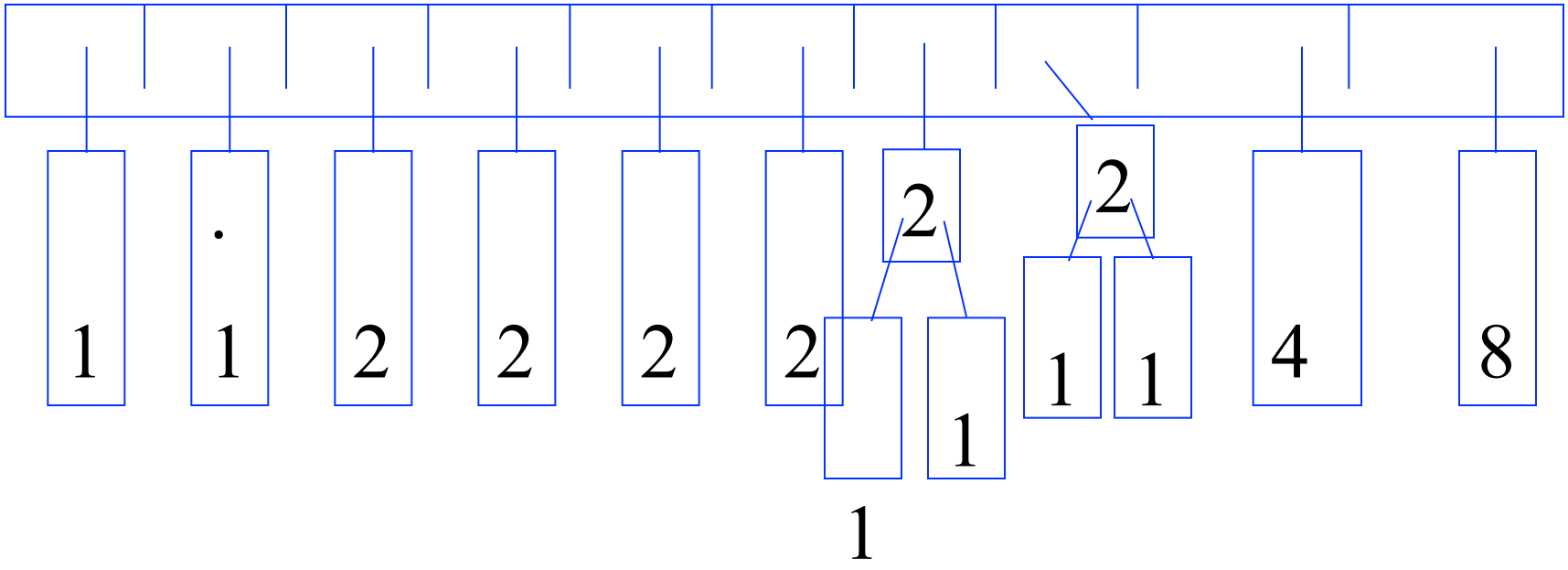

1
1
1
1
1

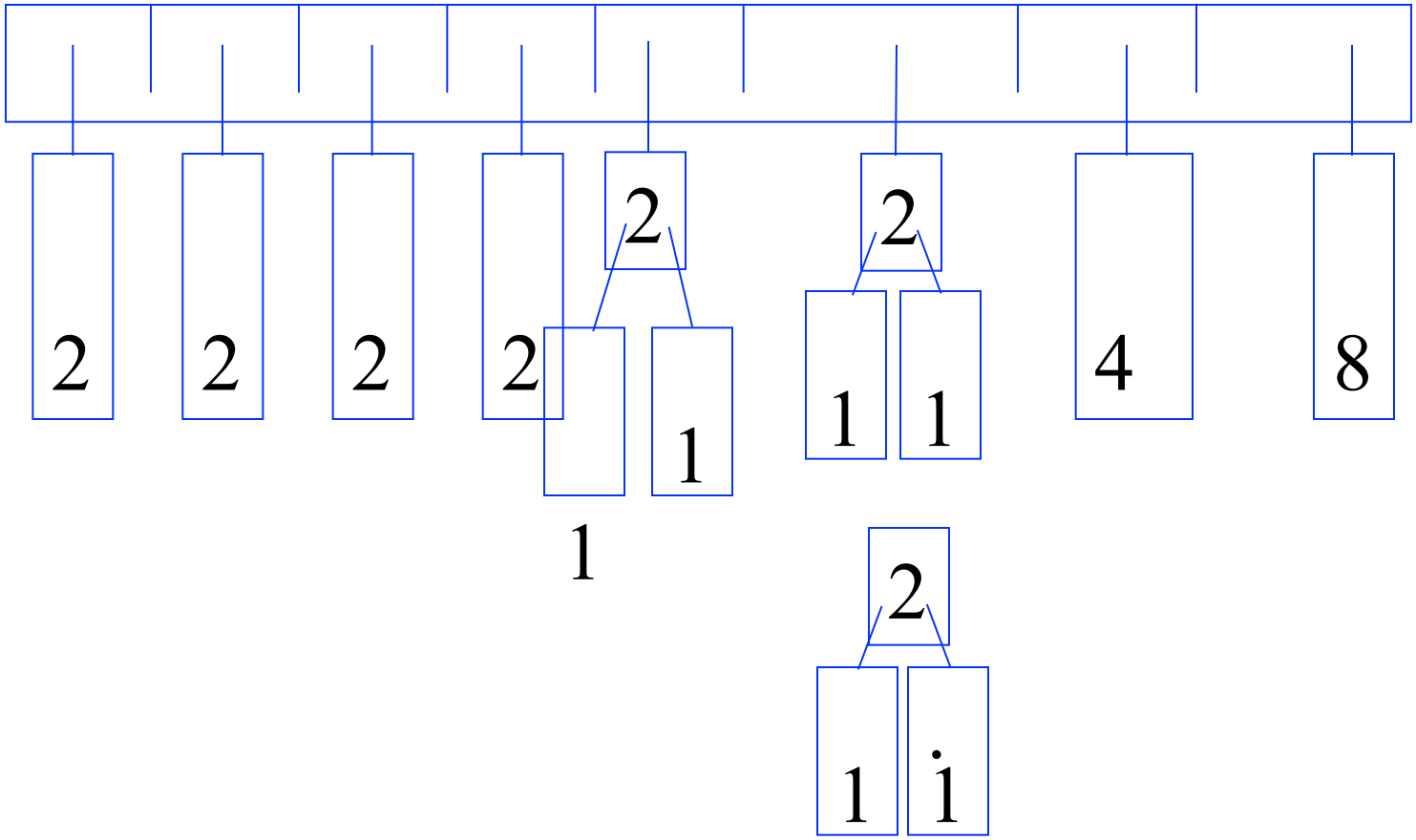

1 1 1 1 1 . 1 2 2 2 2 4 8

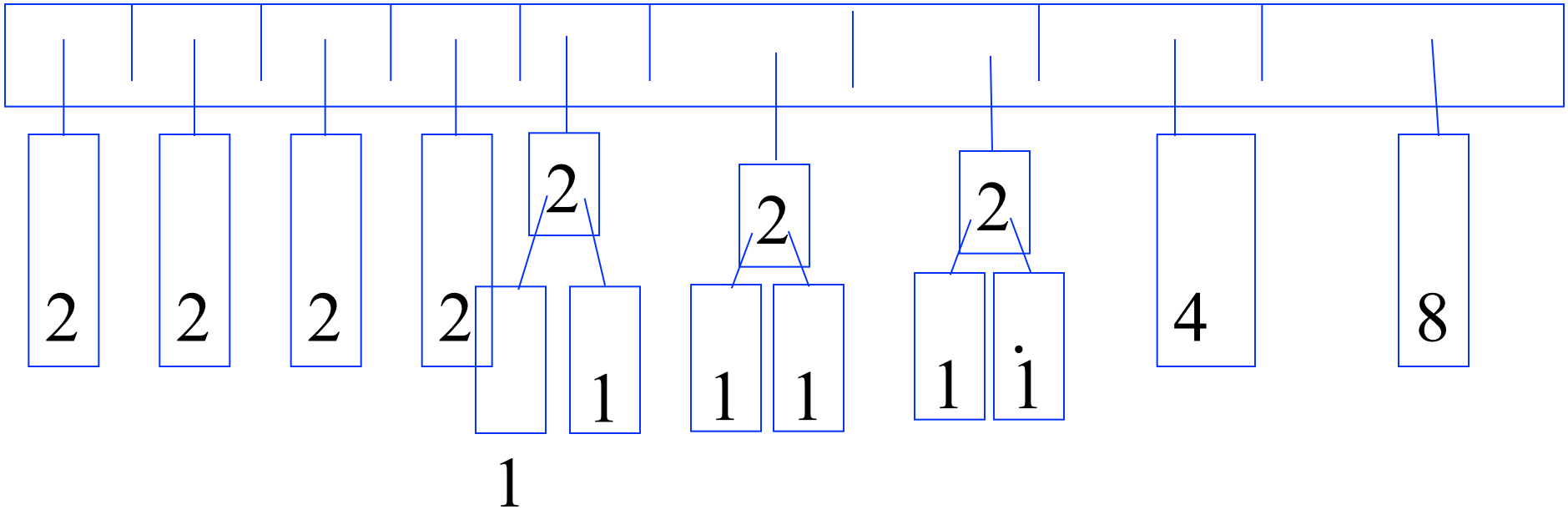


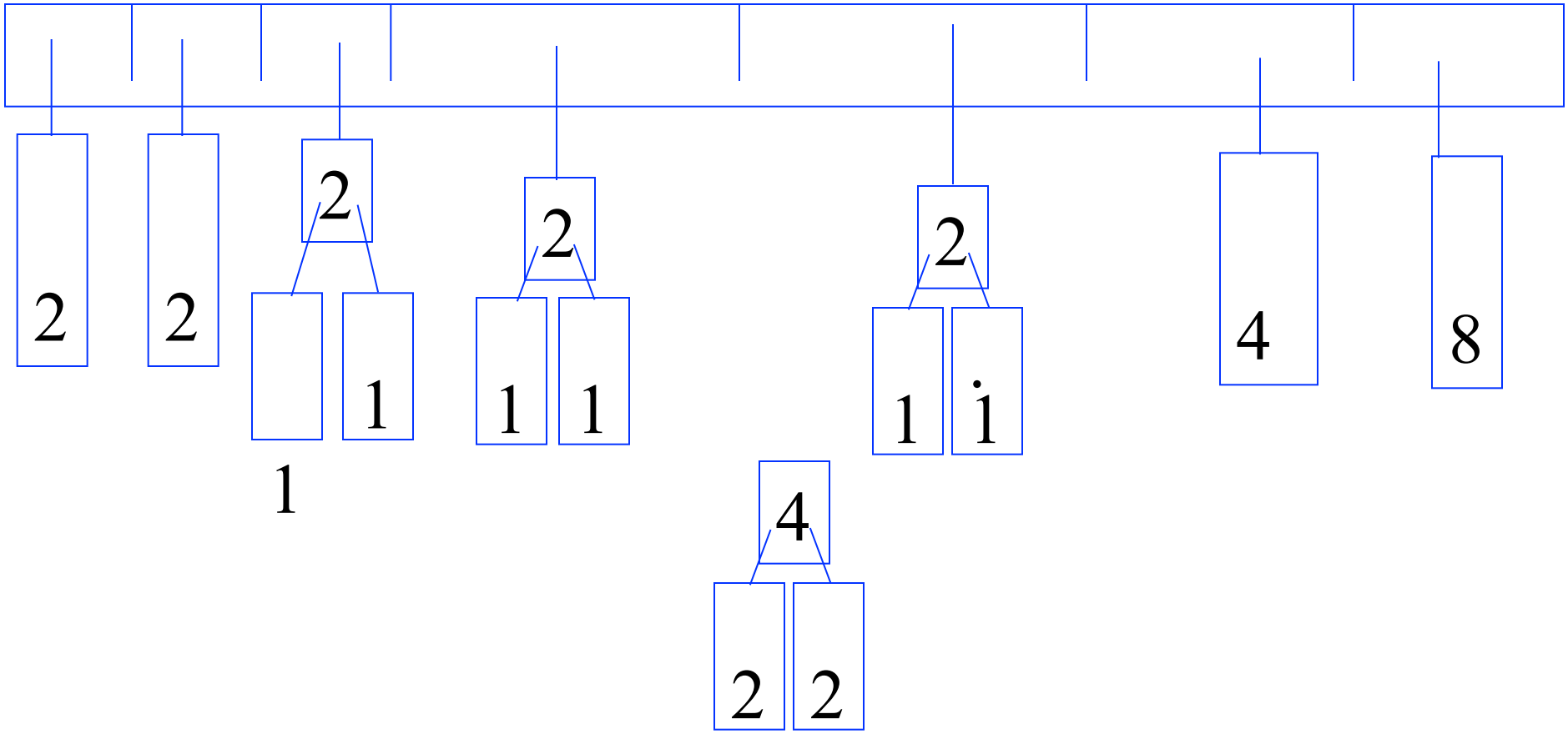

1
1
1
.
1
2
2
2
2
2
1
1
4
8

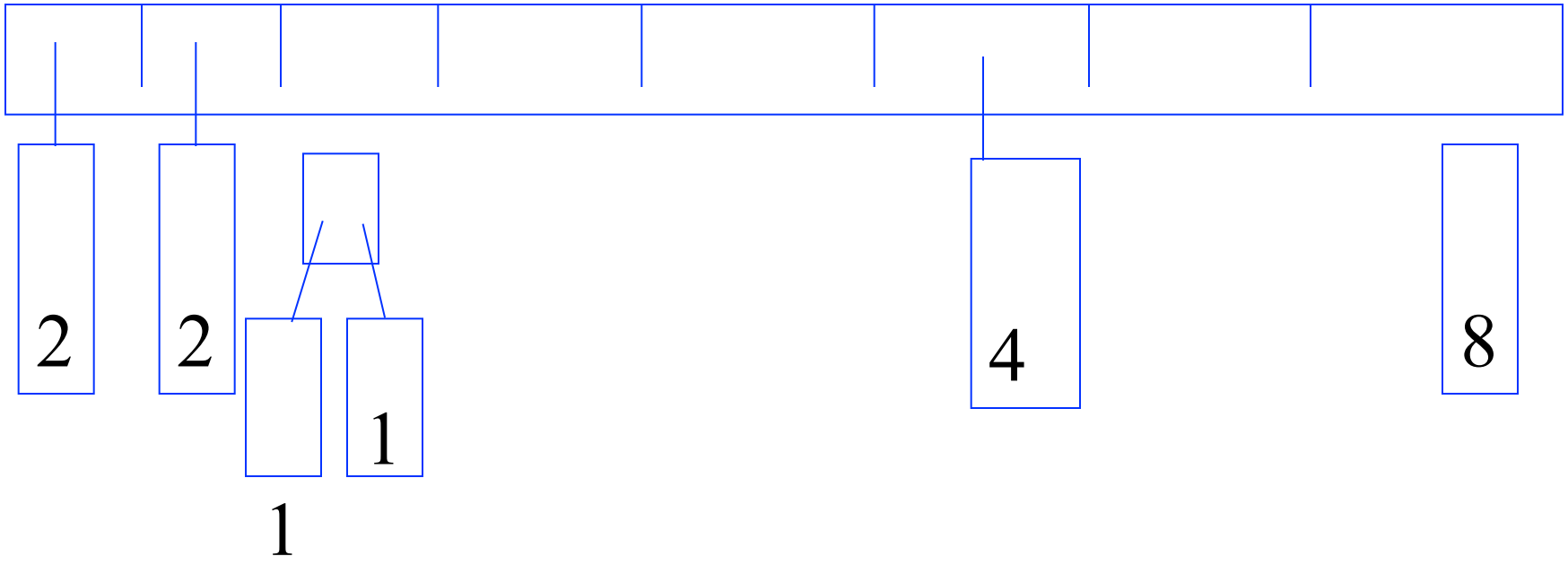

1
.
1
2
2
2
2
2
4
8
1
1
2
1
1

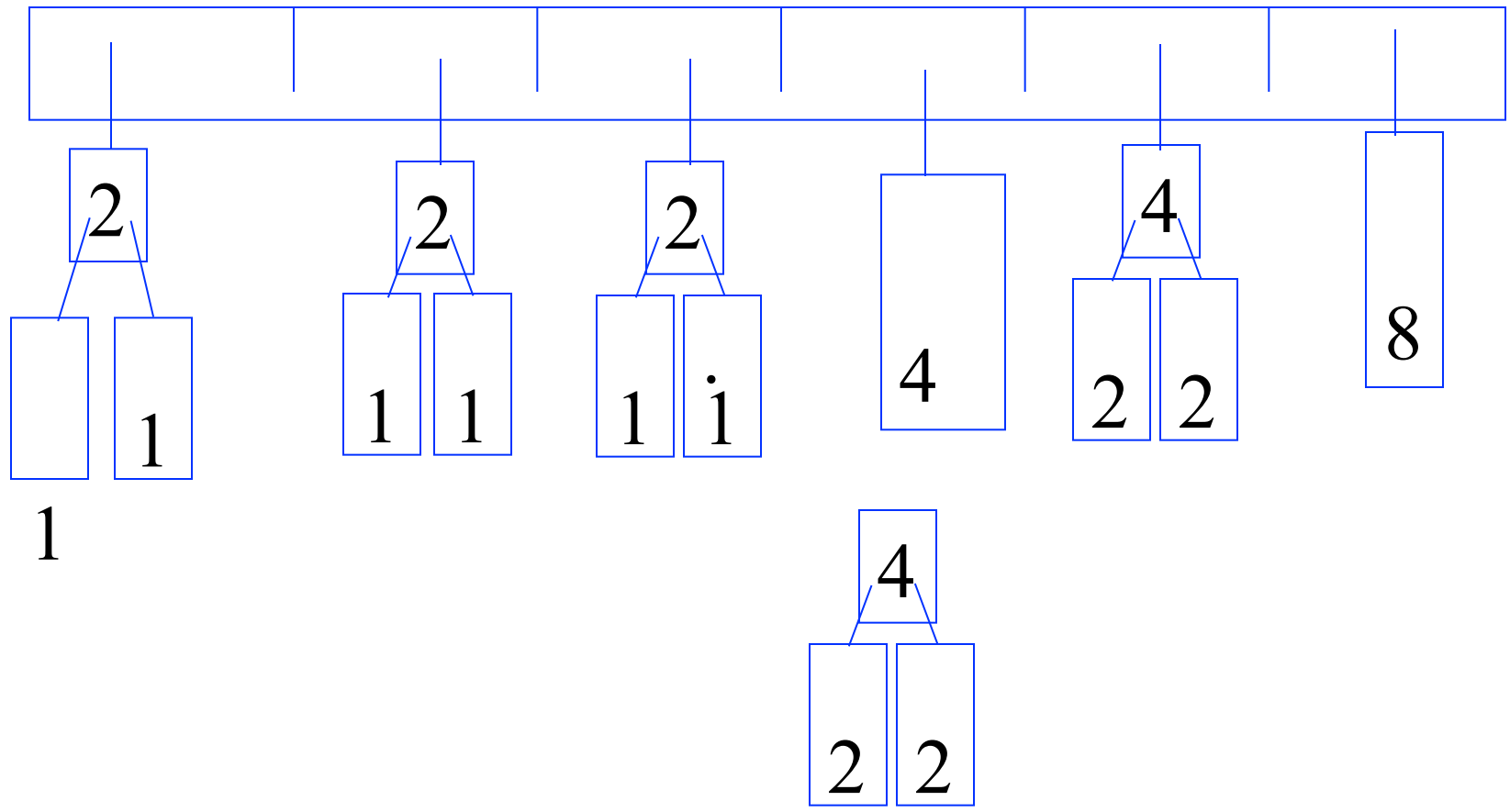

1
.
1
2
2
2
2
2
2
1
1
1
2
1
4
8
1

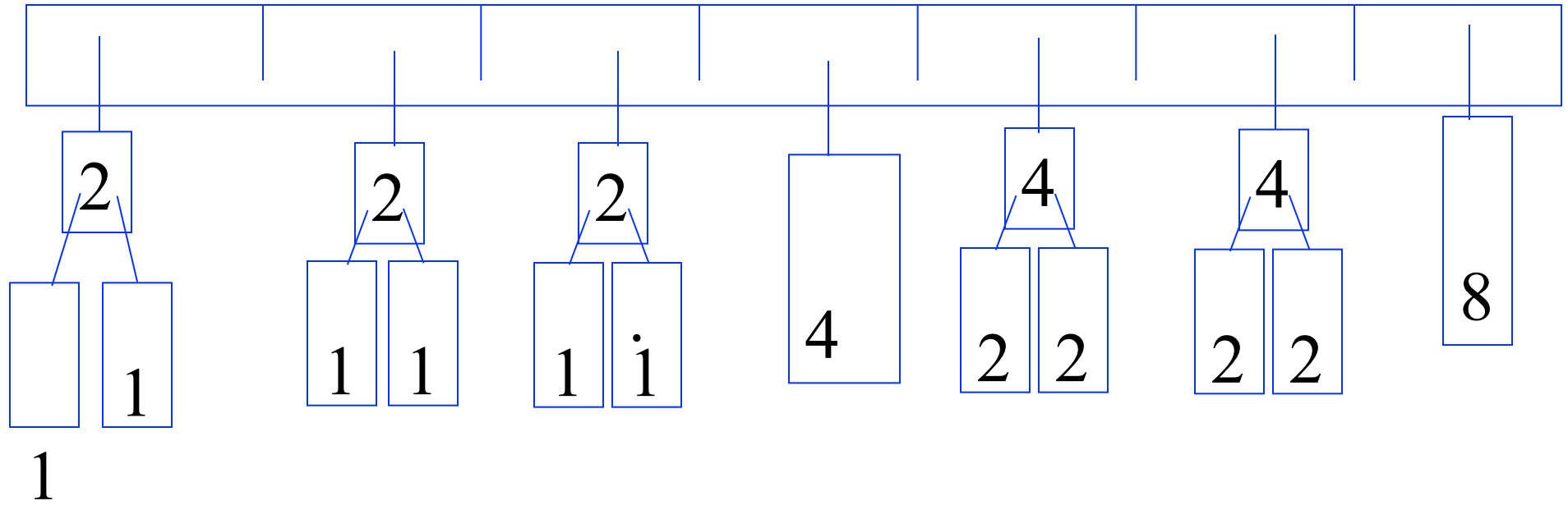

2
2
2
2
2
1
1
2
2
1
1
4
8
2
1
i

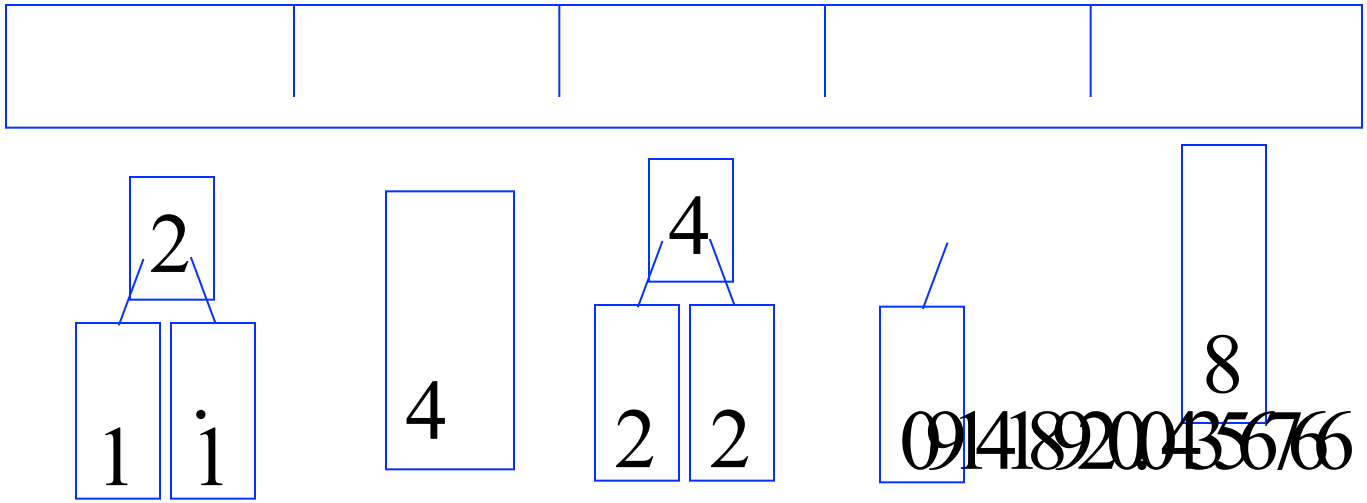

2
2
2
2
2
1
1
2
1
1
2
1
1
4
8

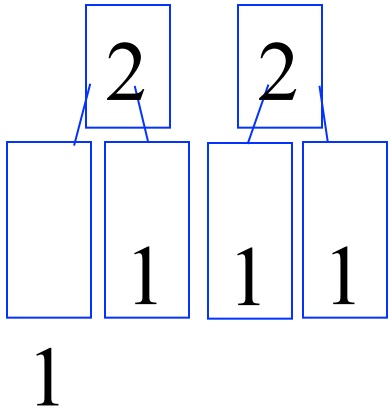

2
2
2
1
1
2
1
1
2
1
i
4
8
4
2
2

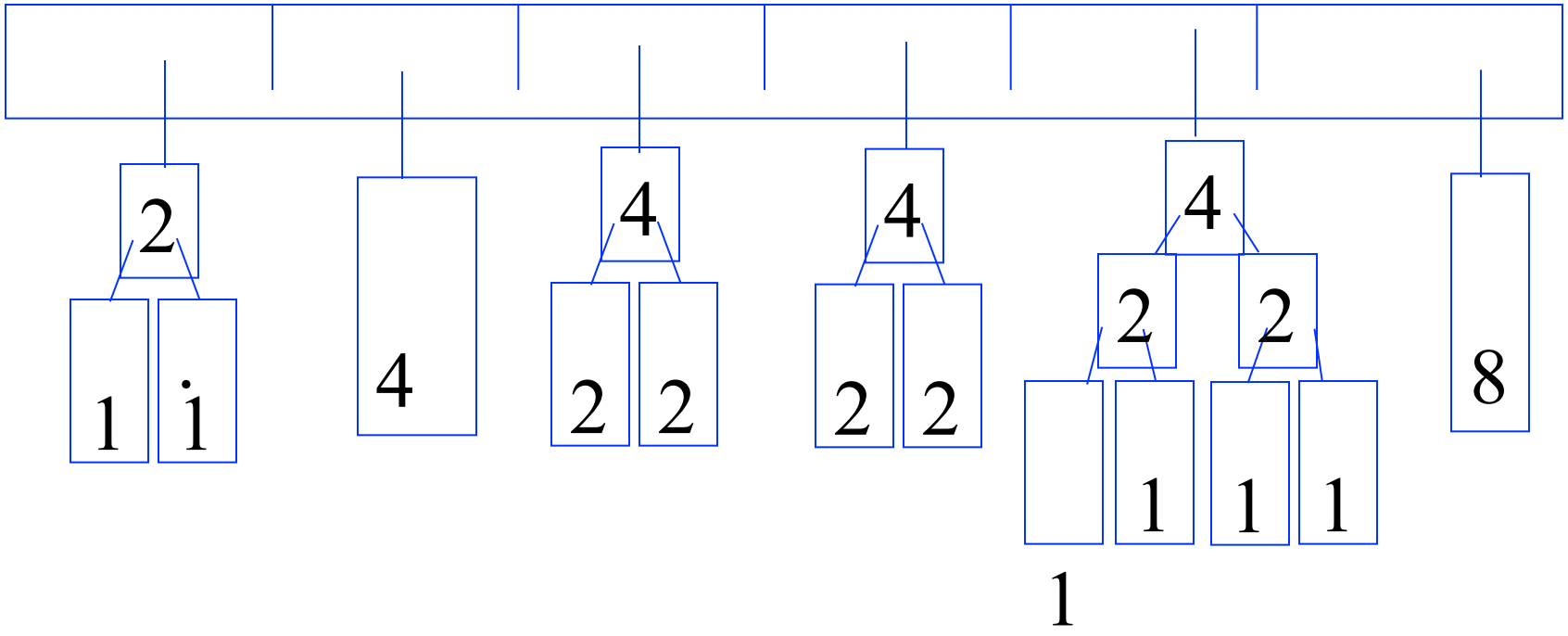

2
2
1
1
4
8

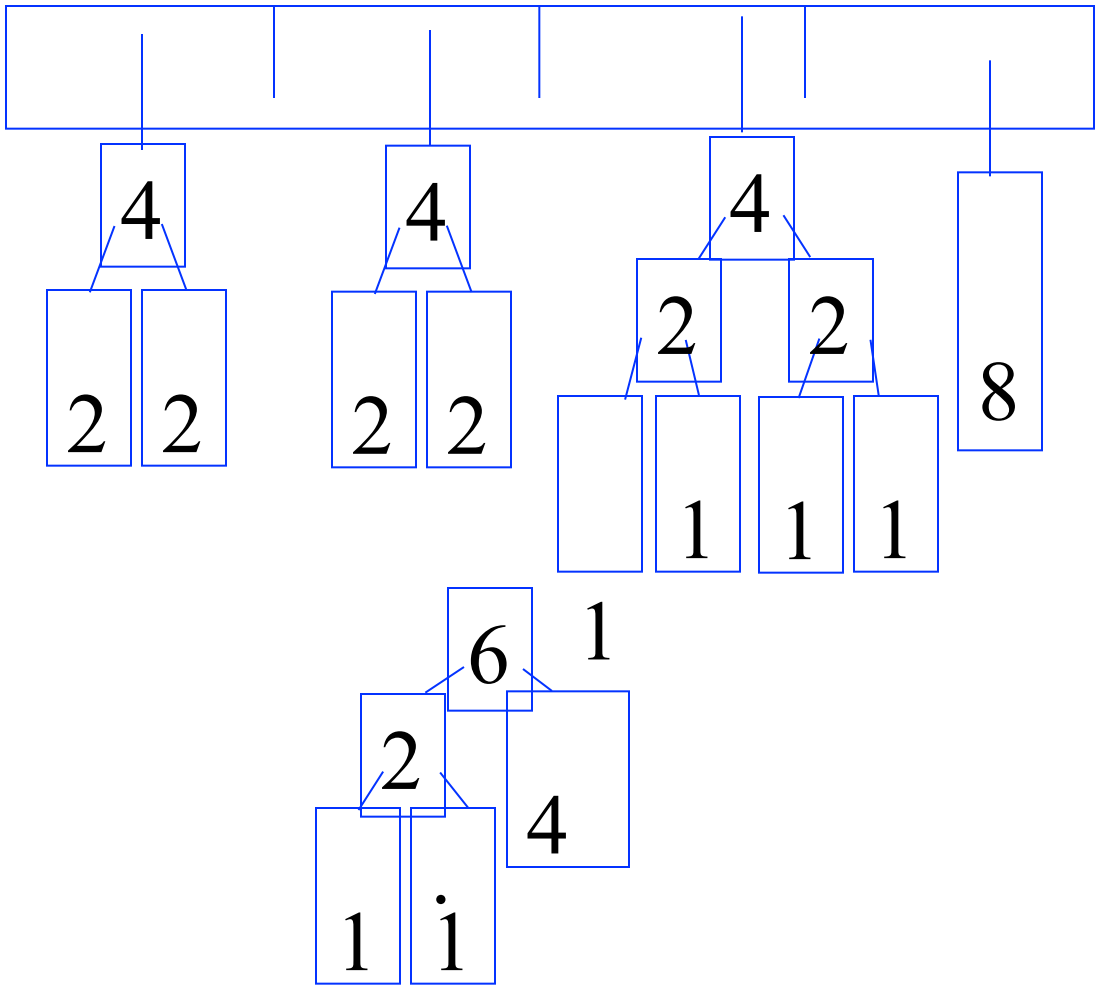

2
2
2
4
4
8
1
1
1
1
1
1
2
2
4
2
2

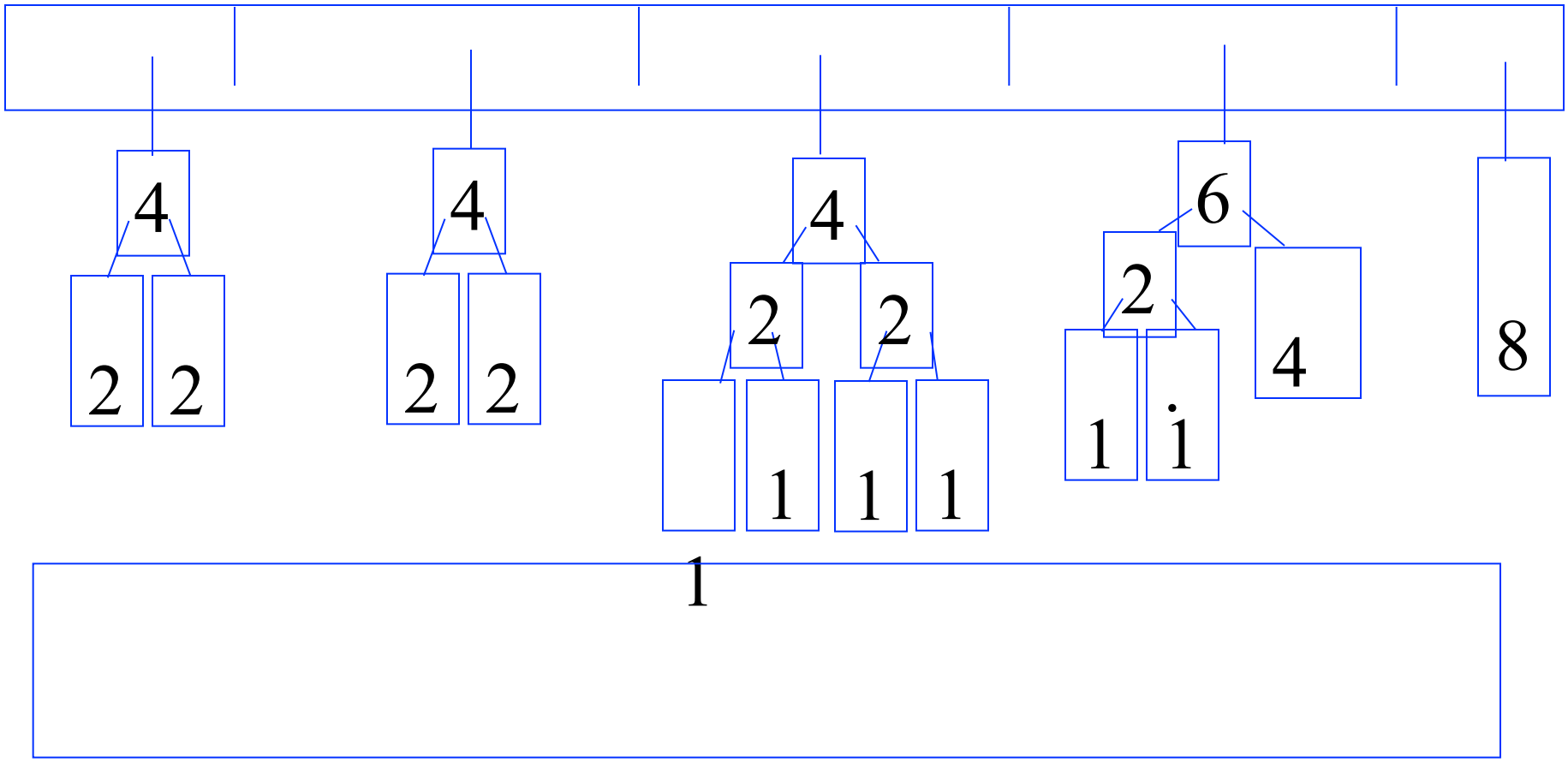

2
2
2
4
4
4
8
1
1
1
1
4
2
2
2
2
1

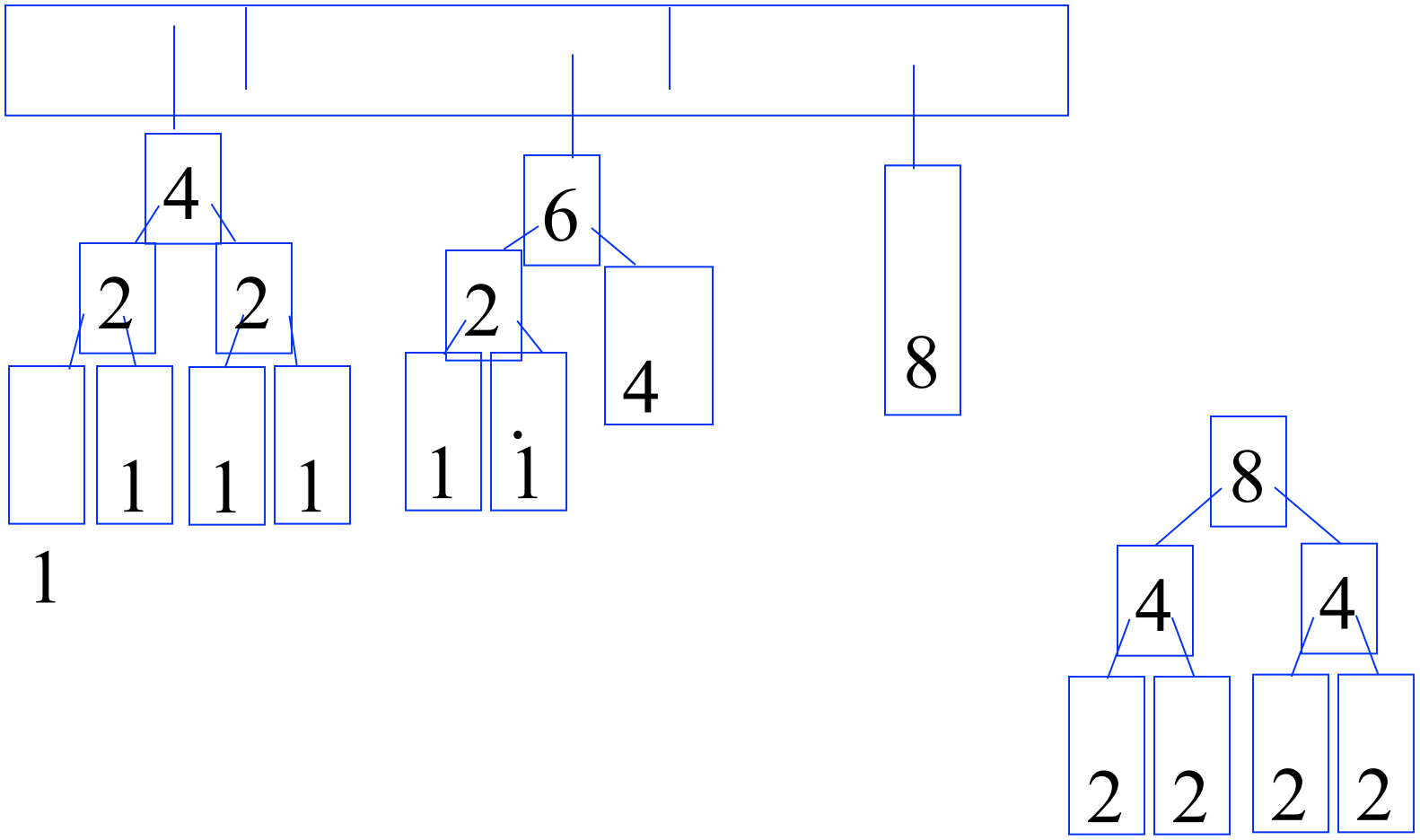

2
1
1
4
4
2
2
8
0
9
1
4
1
8
9
2
0
0
4
3
5
6
7
6
6
1
2
0
1
4
8
0
0
-
0
.
0
0
3
3
7
5
1
2

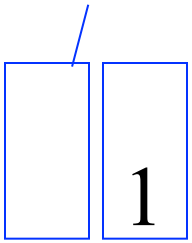

2
2
1
1
1
1


2
1
1
4
4
2
2
4
2
2
4
2
2
1
1
1
8
1

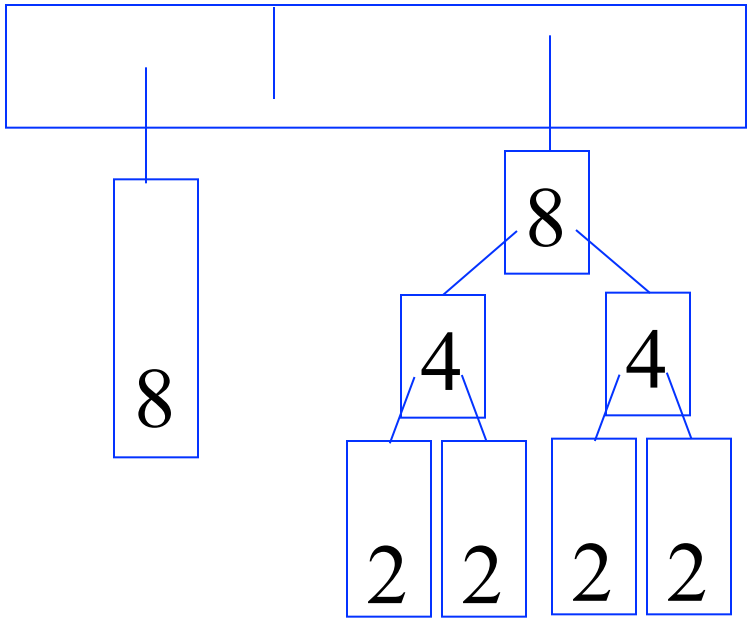

4
4
4
2
2
2
2
2
2
8
1
1
1
6
1
2
4
1
i

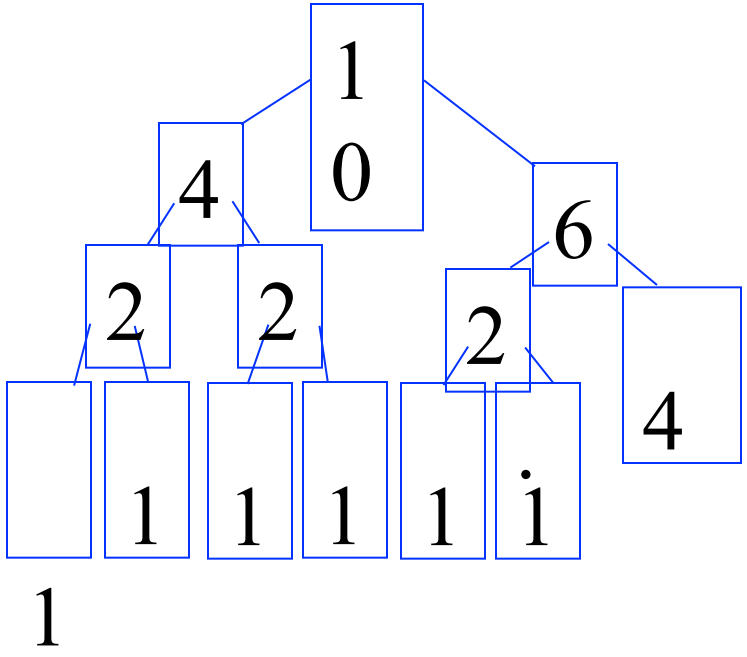

4
2 2
4
2 2
4
2 2
1 1 1
6
2
1 1
4
8
1

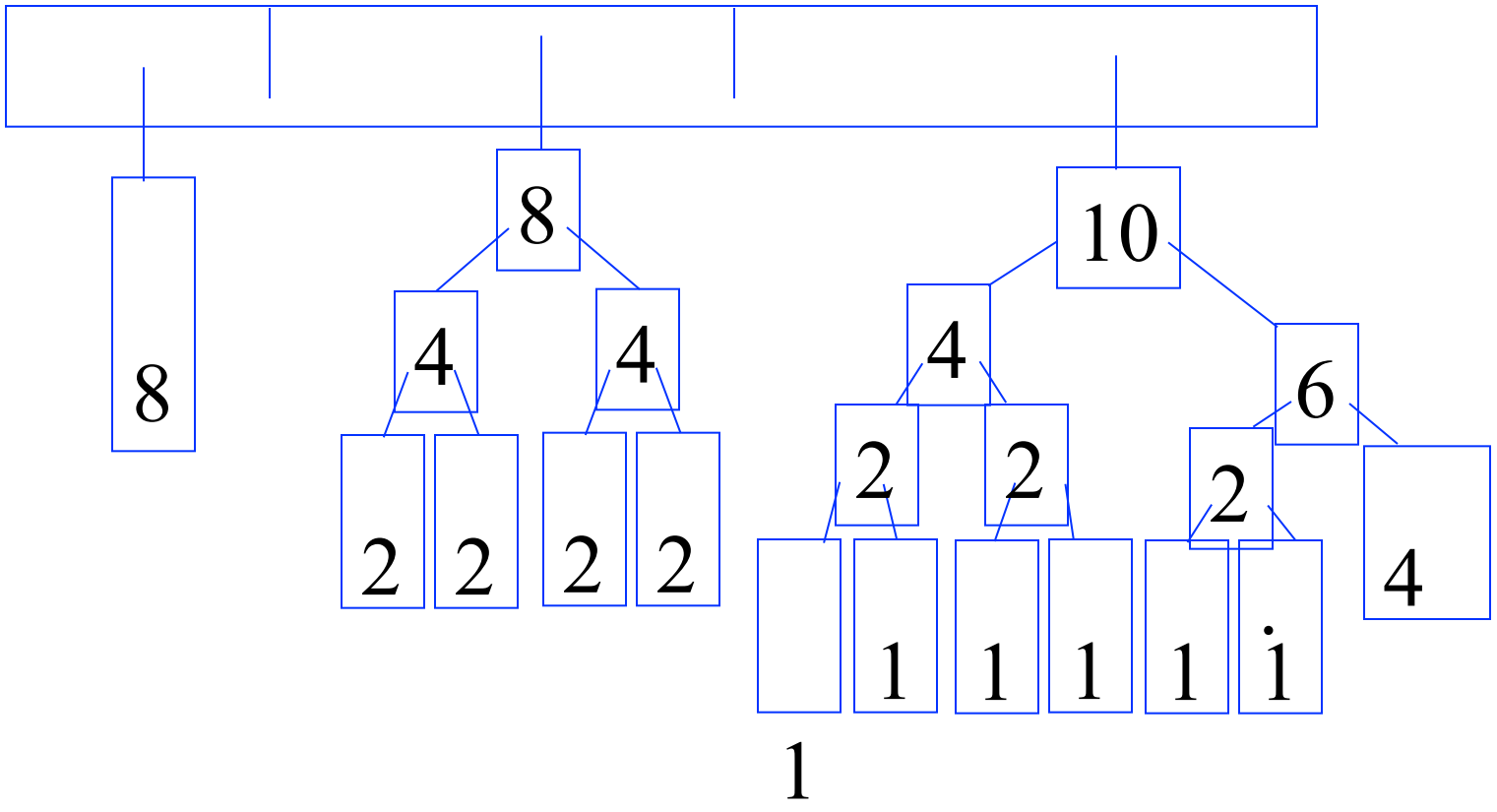

4
2
2
1
1
1
6
2
4
1
1
8
8
4
4
2
2
2
2

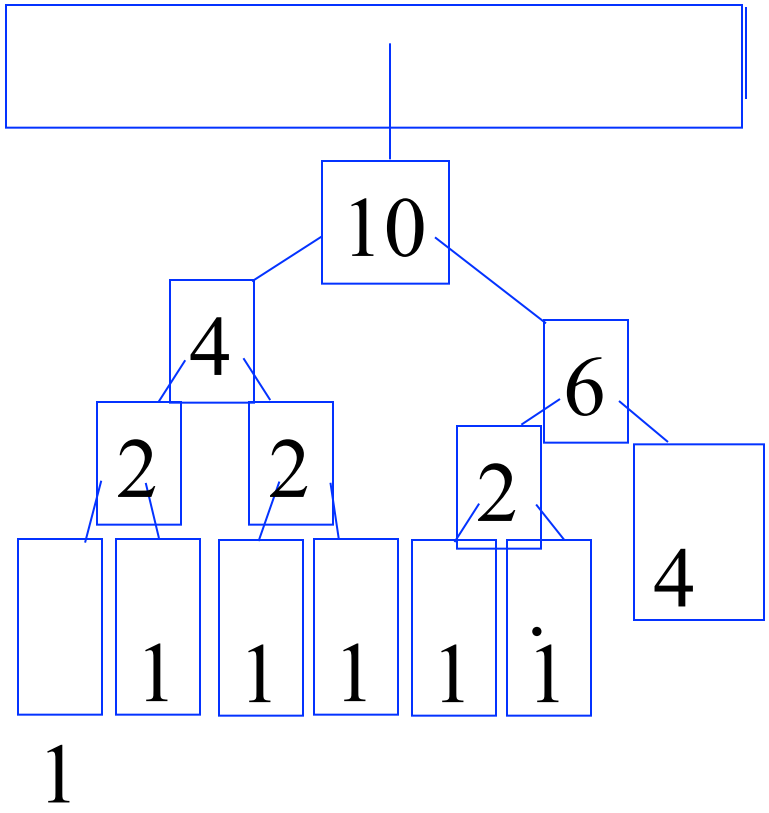


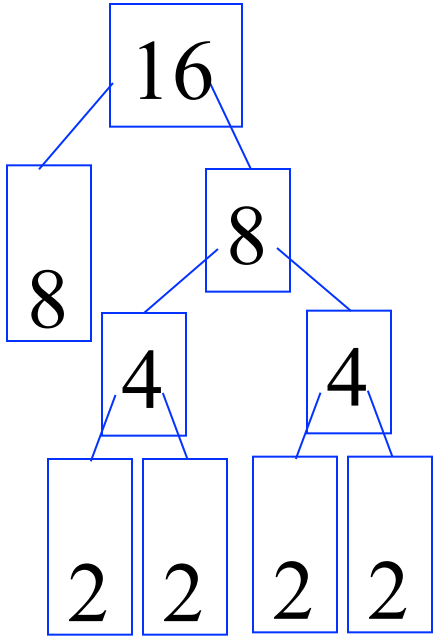
1

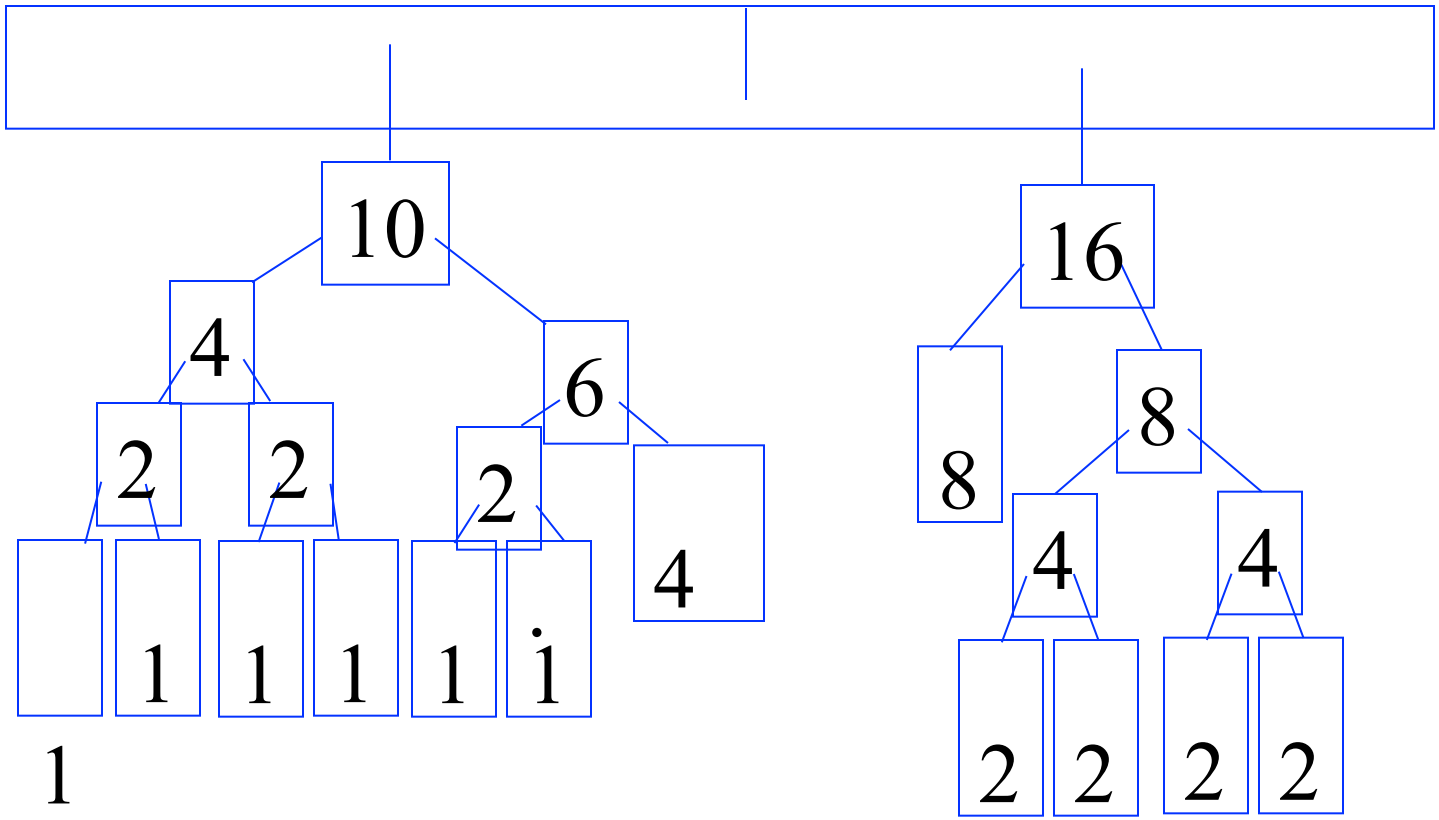

8
8
4
4
2
2
2
2

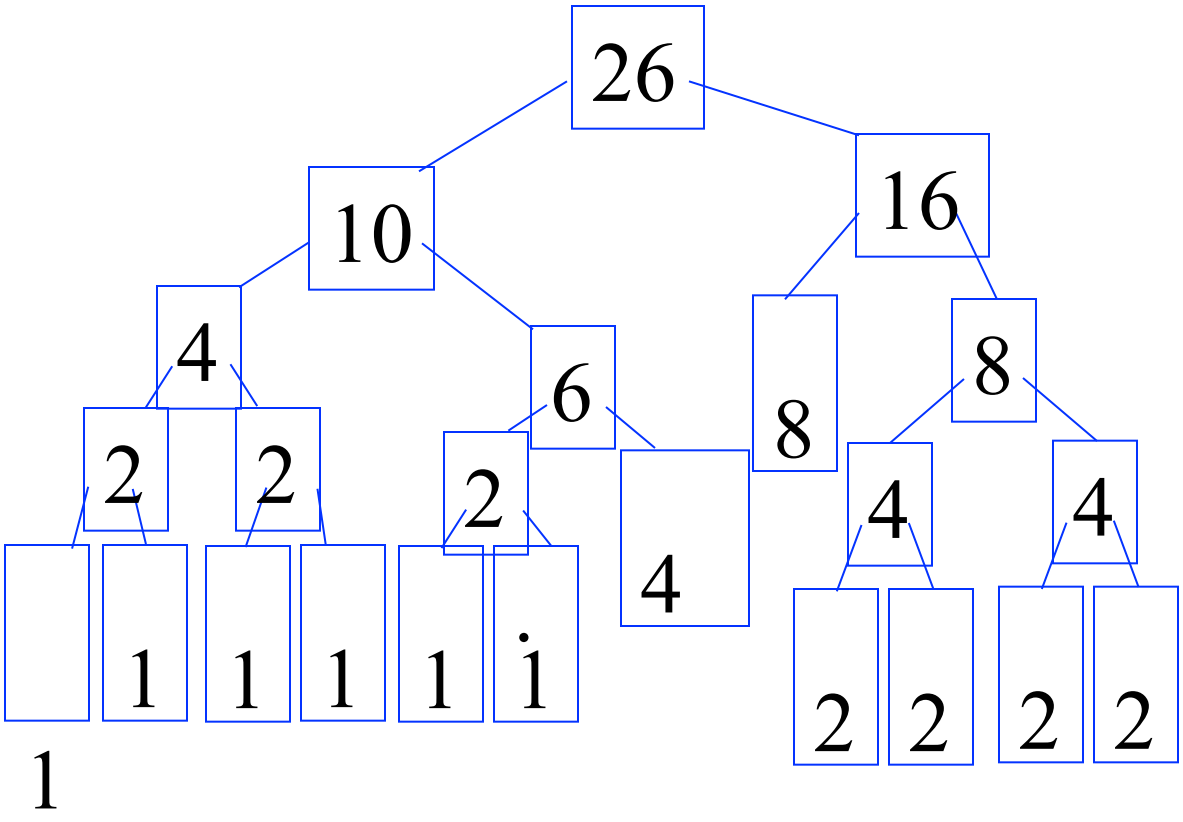

1
0
4
6
2
2
2
4
1
1
1
1
i
1

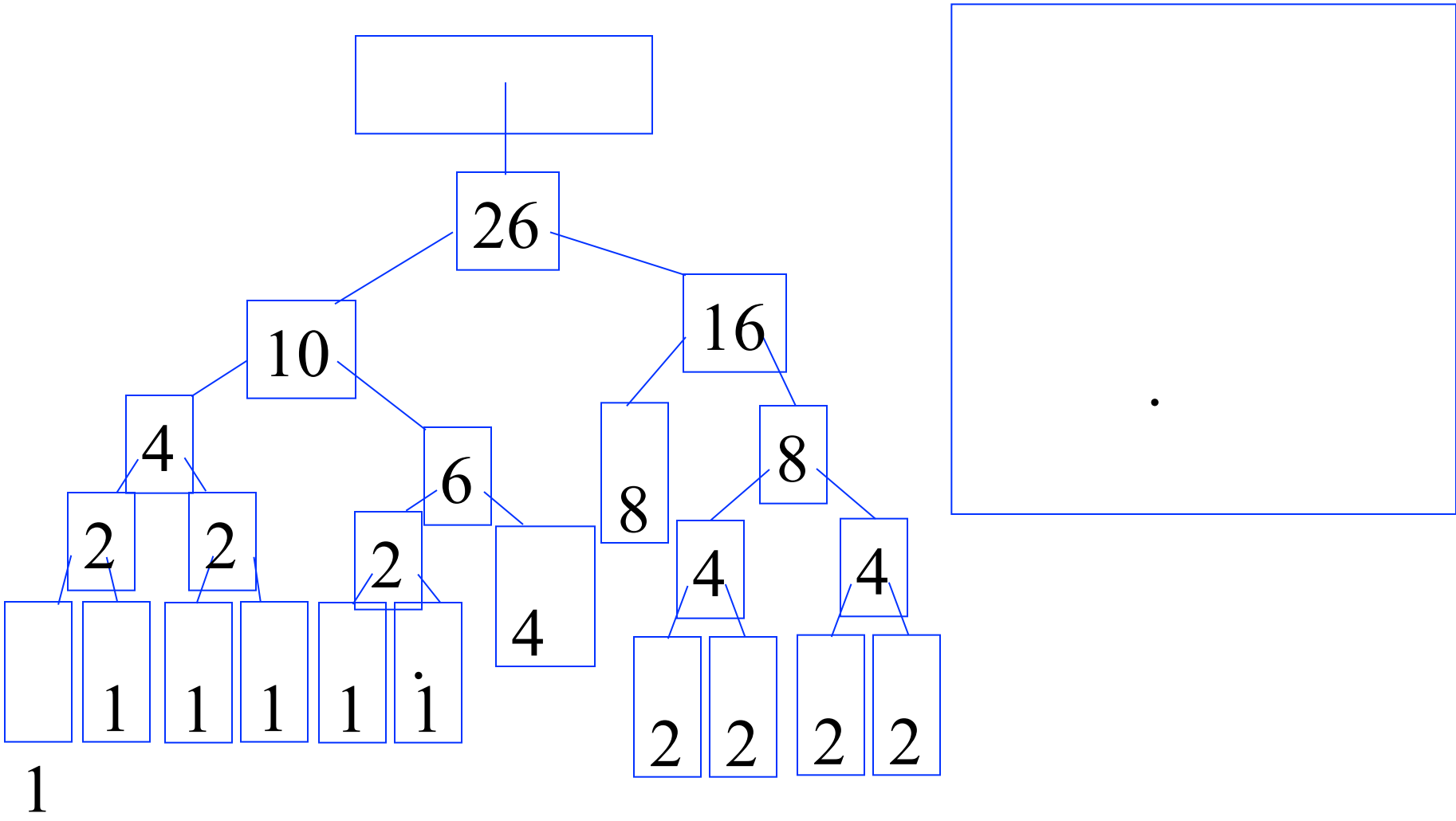

8
8
4
4
2
2
2
2
10
4
2
2
6
2
4
1
1
1
1
1
1

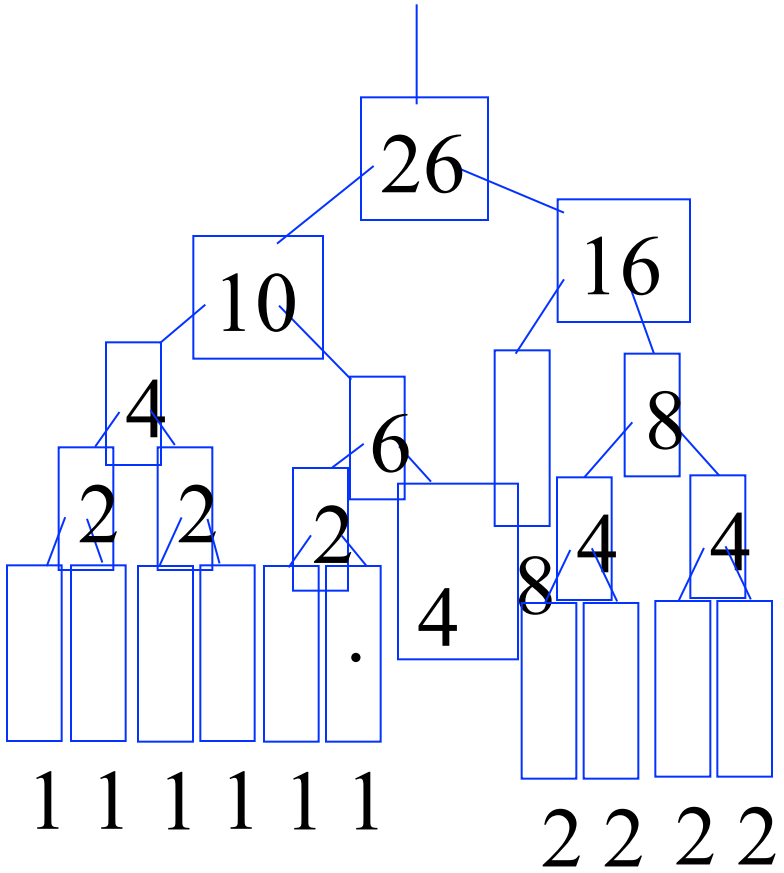

10
4
6
2
2
2
4
1
1
1
1
1
1


16
8
8
4
4
2
2
2
2


10
4
6
2
2
2
4
1
1
1
1
1
1
16
8
8
4
4
2
2
2
2


26
10
16
4
6
8
8
2
2
2
4
4
4
1
1
1
1
1
2
2
2
2
1


26
10
16
4
6
8
8
2
2
2
4
4
4
1
1
1
1
1
2
2
2
2
1



✓ 26

/

.

- **Exercises: P316-3**