

طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده
استادیار گروه مهندسی کامپیوتر دانشگاه قم

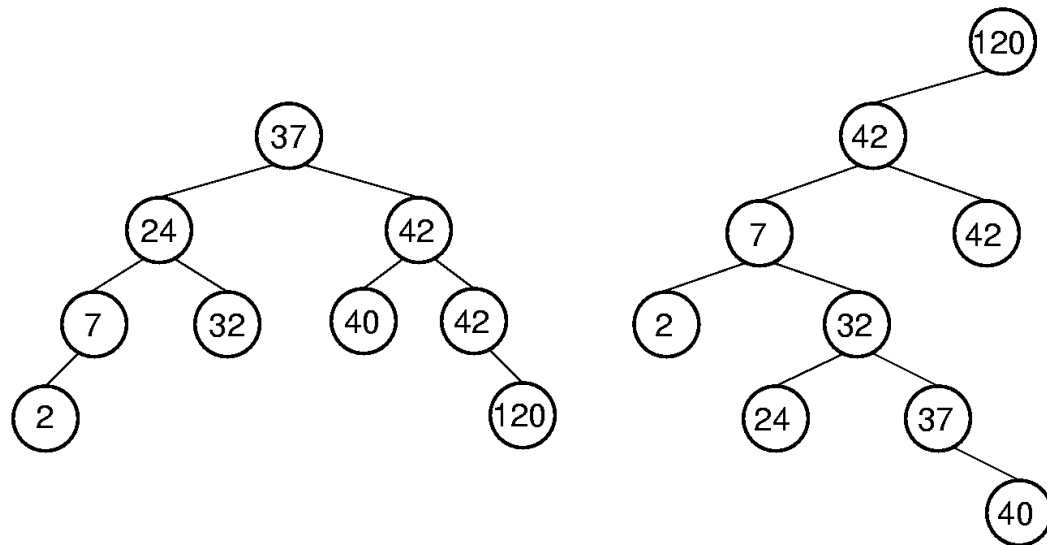
درخت جستجوی دودویی

Binary Search Tree (BST)

BST Property:

All elements stored in the left subtree of a node with value K have values $< K$.

All elements stored in the right subtree of a node with value K have values $\geq K$.



درخت جستجوی دودویی

BST توابع { *search*
Minimum
maximum
predecessor
successor
Insert
Delete } → θ (عمق درخت)

درخت جستجوی دودویی

INORDER-TREE-WALK(x)

```
1  if  $x \neq \text{NIL}$ 
2      then INORDER-TREE-WALK( $\text{left}[x]$ )
3           print  $\text{key}[x]$ 
4           INORDER-TREE-WALK( $\text{right}[x]$ )
```

درخت جستجوی دودویی

TREE-SEARCH(x, k)

```
1  if  $x = \text{NIL}$  or  $k = \text{key}[x]$ 
2    then return  $x$ 
3  if  $k < \text{key}[x]$ 
4    then return TREE-SEARCH( $\text{left}[x], k$ )
5    else return TREE-SEARCH( $\text{right}[x], k$ )
```

درخت جستجوی دودویی

ITERATIVE-TREE-SEARCH(x, k)

```
1  while  $x \neq \text{NIL}$  and  $k \neq \text{key}[x]$ 
2      do if  $k < \text{key}[x]$ 
3          then  $x \leftarrow \text{left}[x]$ 
4          else  $x \leftarrow \text{right}[x]$ 
5  return  $x$ 
```

درخت جستجوی دودویی

TREE-MINIMUM(x)

```
1  while  $left[x] \neq NIL$ 
2      do  $x \leftarrow left[x]$ 
3  return  $x$ 
```

TREE-MAXIMUM(x)

```
1  while  $right[x] \neq NIL$ 
2      do  $x \leftarrow right[x]$ 
3  return  $x$ 
```

درخت جستجوی دودویی

TREE-SUCCESSOR(x)

```
1  if  $right[x] \neq \text{NIL}$ 
2      then return TREE-MINIMUM( $right[x]$ )
3   $y \leftarrow p[x]$ 
4  while  $y \neq \text{NIL}$  and  $x = right[y]$ 
5      do  $x \leftarrow y$ 
6       $y \leftarrow p[y]$ 
7  return  $y$ 
```


درخت جستجوی دودویی

Theorem 12.2

The dynamic-set operations SEARCH, MINIMUM, MAXIMUM, SUCCESSOR, and PREDECESSOR can be made to run in $O(h)$ time on a binary search tree of height h . ■

درخت جستجوی دودویی

12.2-2

Write recursive versions of the TREE-MINIMUM and TREE-MAXIMUM procedures.

12.2-3

Write the TREE-PREDECESSOR procedure.

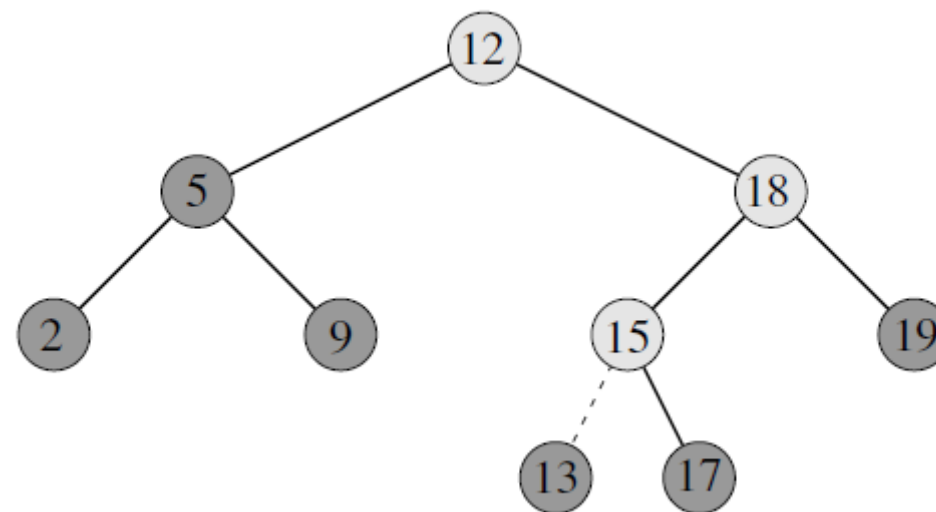
درخت جستجوی دودویی

TREE-INSERT(T, z)

```

1   $y \leftarrow \text{NIL}$ 
2   $x \leftarrow \text{root}[T]$ 
3  while  $x \neq \text{NIL}$ 
4      do  $y \leftarrow x$ 
5          if  $\text{key}[z] < \text{key}[x]$ 
6              then  $x \leftarrow \text{left}[x]$ 
7              else  $x \leftarrow \text{right}[x]$ 
8   $p[z] \leftarrow y$ 
9  if  $y = \text{NIL}$ 
10     then  $\text{root}[T] \leftarrow z$ 
11     else if  $\text{key}[z] < \text{key}[y]$ 
12         then  $\text{left}[y] \leftarrow z$ 
13         else  $\text{right}[y] \leftarrow z$ 

```



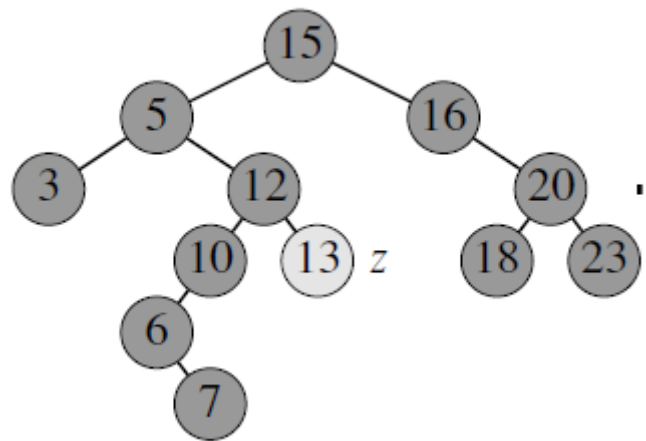
▷ Tree T was empty

درخت جستجوی دودویی

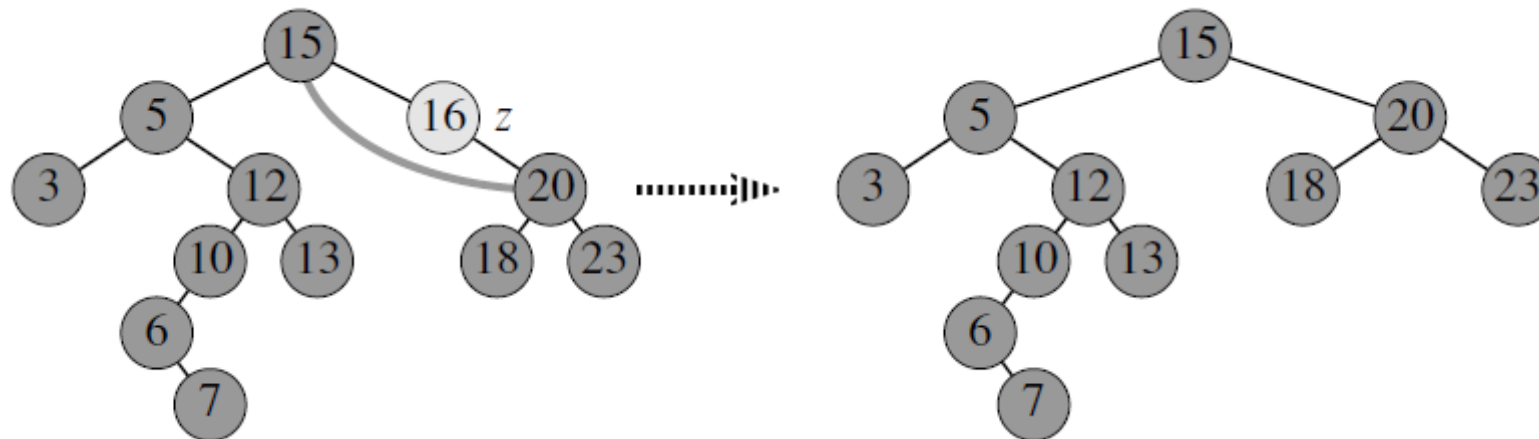
TREE-DELETE(T, z)

```
1  if  $left[z] = \text{NIL}$  or  $right[z] = \text{NIL}$ 
2    then  $y \leftarrow z$ 
3    else  $y \leftarrow \text{TREE-SUCCESSOR}(z)$ 
4  if  $left[y] \neq \text{NIL}$ 
5    then  $x \leftarrow left[y]$ 
6    else  $x \leftarrow right[y]$ 
7  if  $x \neq \text{NIL}$ 
8    then  $p[x] \leftarrow p[y]$ 
9  if  $p[y] = \text{NIL}$ 
10   then  $root[T] \leftarrow x$ 
11   else if  $y = left[p[y]]$ 
12         then  $left[p[y]] \leftarrow x$ 
13         else  $right[p[y]] \leftarrow x$ 
14  if  $y \neq z$ 
15    then  $key[z] \leftarrow key[y]$ 
16         copy  $y$ 's satellite data into  $z$ 
17  return  $y$ 
```

درخت جستجوی دودویی



درخت جستجوی دودویی



درخت جستجوی دودویی

