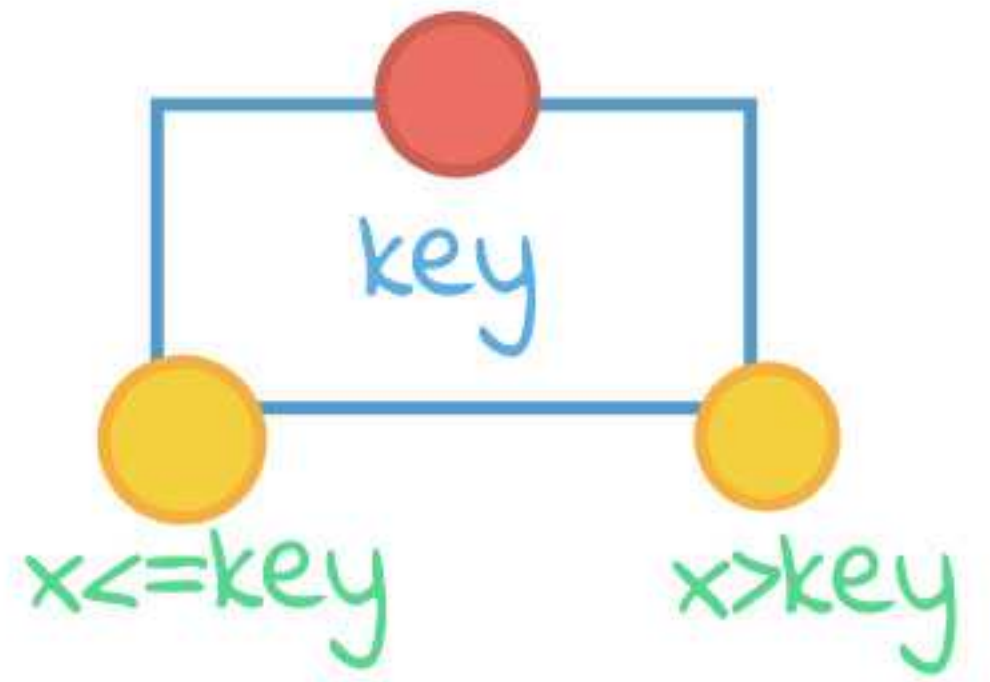


```
0 class BNode {
1   BNode parent;
2   int key;
3   BNode left;
4   BNode right;
5   BNode(int k) { key=k; }
6 }
7
```



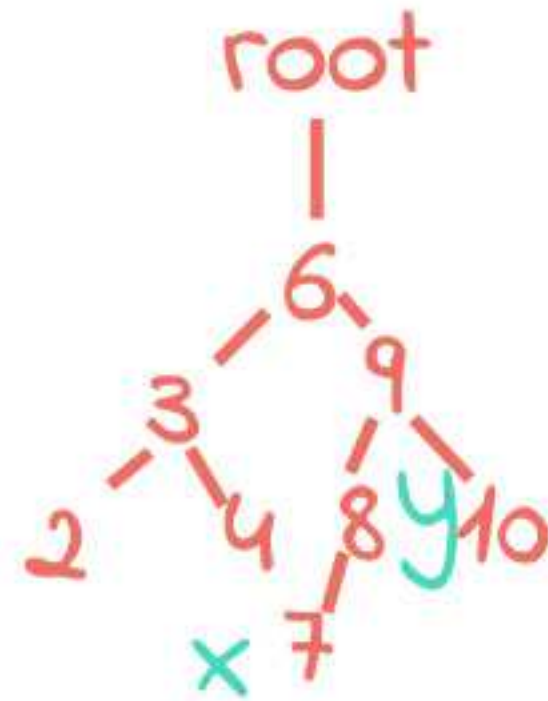
```
9 class BTree {
10 BNode root;
10 ... insert, minimum, successor, delete
10 }
```

```

0
1 public void insert(int k)
2   { BNode z = new BNode(k);
3     BNode y = null;
4     BNode x = root;
5     while (x != null)
6       { y = x;
7         if (k < x.key) x = x.left; else x = x.right;
8       }
9     z.parent = y;
10    if (y == null)
11      { root = z; }
12    else
13      if (k < y.key)
14        { y.left = z; }
15      else
16        { y.right = z; }
17  }

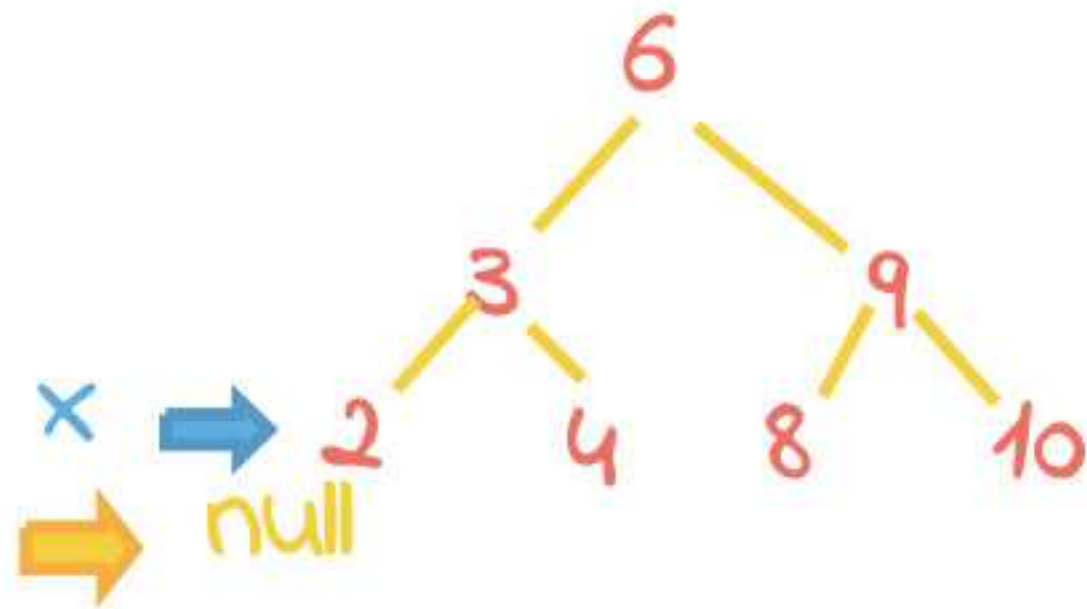
```

→ z =



null

```
0  
1 public BNode minimum(BNode x)  
2   { while (x.left!=null)  
3     { x=x.left; }  
4     return x;  
5   }
```



```
0 public BNode successor(BNode x)
```

```
1 { BNode y;
```

```
2   if (x.right!=null)
```

```
3     return minimum(x.right);
```

```
4   else { y = x.parent; }
```

```
5   while ((y != null) && (x == y.right))
```

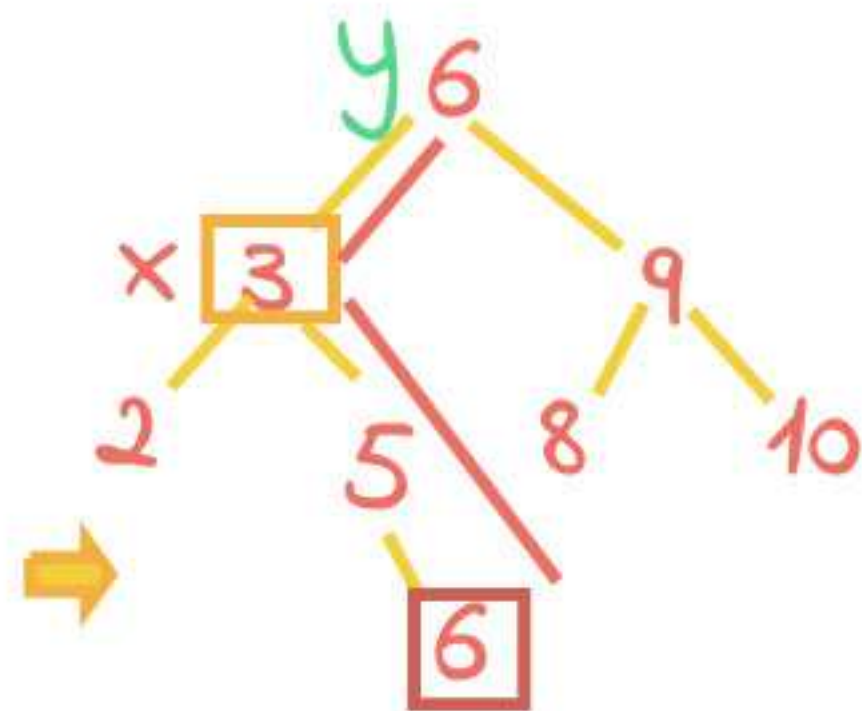
```
6     { x=y;
```

```
7       y=x.parent;
```

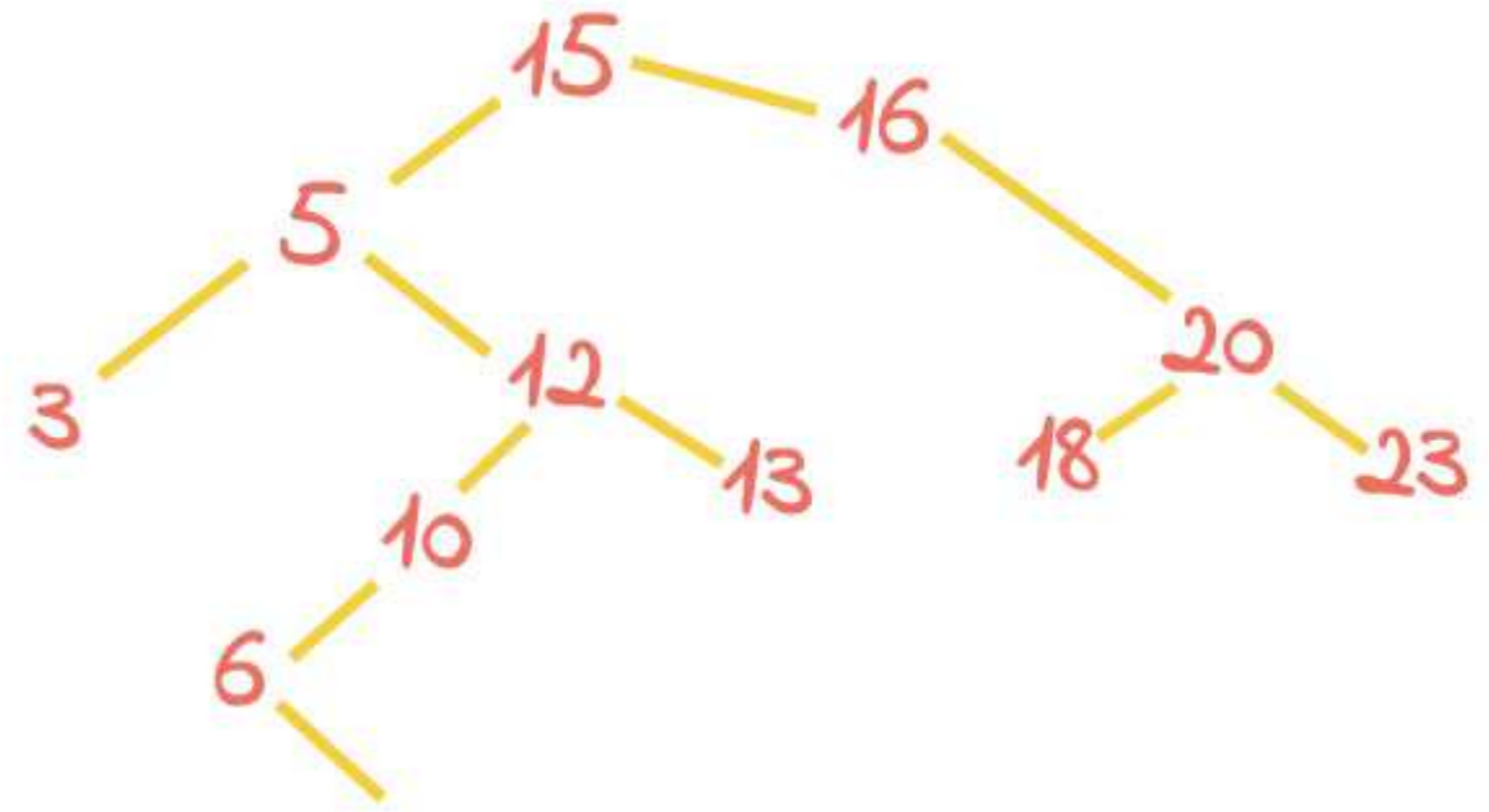
```
8     }
```

```
9   return y;
```

```
10 }
```



x y z

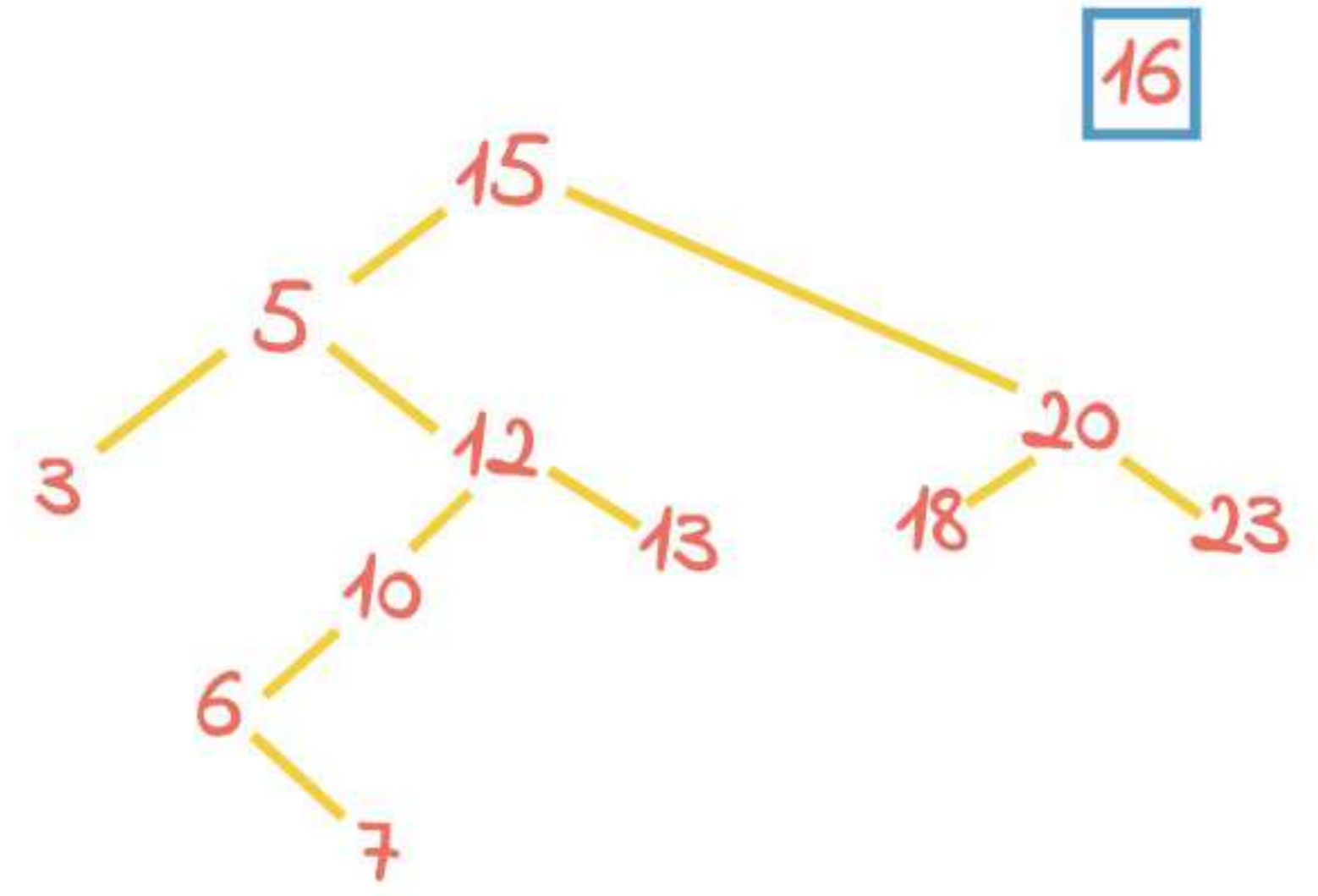


7

```
0 public BNode delete(BNode z)
1 { BNode x,y;
2   if ((z.left==null) || (z.right==null))
3     { y=z; }
4   else
5     { y=successor(z); }
6   if (y.left!=null)
7     { x=y.left; }
8   else
9     { x=y.right; }
10  if (x != null)
11    { x.parent = y.parent; }
12  if ((y.parent==null))
13    { if (y==root) root=x; }
14  else
15    if (y == y.parent.left)
16      { y.parent.left=x; }
17    else { y.parent.right=x; }
18  if (y!=z)
19    { z.key=y.key; }
20  return y;
21 }
22 }
```

x y z

```
0 public BNode delete(BNode z)
1 { BNode x,y;
2   if ((z.left==null) || (z.right==null))
3     { y=z; }
4   else
5     { y=successor(z); }
6   if (y.left!=null)
7     { x=y.left; }
8   else
9     { x=y.right; }
10  if (x != null)
11    { x.parent = y.parent; }
12  if ((y.parent==null))
13    { if (y==root) root=x; }
14  else
15    if (y == y.parent.left)
16      { y.parent.left=x; }
17    else { y.parent.right=x; }
18  if (y!=z)
19    { z.key=y.key; }
20  return y;
21 }
22 }
```



```

0 public BNode delete(BNode z)
1   { BNode x,y;
2     if ((z.left==null) || (z.right==null))
3       { y=z; }
4     else
5       { y=successor(z); }
6     if (y.left!=null)
7       { x=y.left; }
8     else
9       { x=y.right; }
10    if (x != null)
11      { x.parent = y.parent; }
12    if ((y.parent==null))
13      { if (y==root) root=x; }
14    else
15      if (y == y.parent.left)
16        { y.parent.left=x; }
17      else { y.parent.right=x; }
18    if (y!=z)
19      { z.key=y.key; }
20    return y;
21  }
22

```



y

