The goal is to insert the new item into an existing leaf.

This is easy if the leaf has less than $2t - 1$ elements.

$\text{Insert}(T, 15)$

15 can be inserted to this leaf
The hard case is when the leaf has $2t - 1$ elements.

The solution is to perform the following step during the descend in the tree: Before entering a node, if the node has $2t - 1$ elements, perform node splitting on the node in order to reduce the number of elements.
Node splitting

- Let $v$ be a node with $2t - 1$ elements. The node splitting operation on $v$ replaces $v$ with two nodes: one containing the $t - 1$ smallest elements of $v$, and another containing the $t - 1$ largest elements.
- The median element of $v$ is moved to the parent of $v$. If $v$ is the root, a new root is created.
If \( v \) is not the root, the parent of \( v \) has less than \( 2t - 1 \) elements, so it is legal to move the median to the parent.
Example

```
10 30 50
2 4 8
t=2
12 20 30 60 65
Insert(T,9)
```
Example

Split node

Insert(T,9)

$\text{t}=2$

Diagram:

```
    10 30 50
   /    /    /
  2 4 8 12 20 30 60 65
```
Example

```
Example

10
2 4 8
t=2
12 20 30 60 65
Insert(T,9)
30
50
```

```
      30
     /   \
   10    50
  /   \  /   \
2 4 8 12 20 30 60 65
```
Example

Insert(T,9)

Split node

2 4 8 12 20 30 60 65

30

10 50

t=2

Insert(T,9)
Example

```
4,10
12 20 30 60 65
Insert(T,9)
30
50
2 8
t=2
```
Example

9 is added to the leaf

4,10
12 20 30 60 65

Insert(T,9)

30

2 8 9 12 20 30 60 65

9 is added to the leaf

Insert(T,9)

$\text{Insert}(T,9)$

$\text{t}=2$
Deletion

- The easy case is deleting an item from a leaf that has more than \( t - 1 \) elements.
- The hard cases are deleting an item from a leaf that has \( t - 1 \) elements, and deleting an item from an internal node.
- The following step is performed during the descend in the tree: Before entering a node \( v \), if \( v \) has \( t - 1 \) elements, perform shifting or merging in order to increase the number of elements.
If \( v \) has an immediate left sibling \( u \) and \( u \) has at least \( t \) elements:

1. Move the element in the parent of \( v \) that “separates” \( v \) and \( u \) to \( v \).
2. Move the maximum element of \( u \) to the parent of \( v \).
Otherwise, if \( v \) has an immediate right sibling \( w \), and \( w \) has at least \( t \) elements then

1. Move the element of the parent of \( v \) that “separates” \( v \) and \( w \) to \( v \).
2. Move the minimum element of \( w \) to the parent of \( v \).
If $v$ has no immediate sibling with at least $t$ elements:

1. Merge $v$ with an immediate sibling.
2. Move the “separating element” from the parent of $v$ to the new node.
Deleting an item

When reaching a node $x$ that contains the element $k$ we want to delete there are several cases:

- If $x$ is a leaf, delete the element.
- If $x$ is an internal node:
  - If the child $y$ of $x$ that contains the predecessor of $k$ has at least $t$ elements, delete the predecessor and put it in $x$ instead of $k$.
  - If the child $z$ of $x$ that contains the successor of $k$ has at least $t$ elements, delete the successor and put it in $x$ instead of $k$.
  - Otherwise, merge $y$ and $z$, and recursively delete $k$ from the merged node.
Example

```
t=3
delete(T,6)
```
delete(T, 6)

6 < 14 so the next node is the 1st child of the current node. This child has 3 > t − 1 elements, so no modification is needed.
Example

3 < 6 < 7 so the next node is the 2nd child of the current node. This child has $3 > t - 1$ elements, so no modification is needed.
Example

t=3
delete(T,6)
Example

The element 6 is deleted from the current node.
Example

t = 3

delete(T,11)
Example

11 < 14 so the next node is the 1st child of the current node. This child has $3 > t - 1$ elements, so no modification is needed.
11 is the 3rd item of the node.
The 4th child has $2 = t - 1$ elements, so we can’t replace 11 with its successor.
The 3rd child has $3 > t - 1$ so we replace 11 by its predecessor and delete the predecessor.
Example

\[ t = 3 \]

\[ \text{delete}(T, 11) \]

```
18 21
8 9 10 15 16 17 19 20 22 23 12 13 4 5 1 2
3 7 11
14
```

Diagram:
```
   14
  /   \
3 7 11 /   \
/     /
1 2    4 5 8 9 10 12 13 15 16 17 19 20 22 23
```

Red arrow points to node 11.
The element 10 is deleted from the current node.
The element 11 is replaced by 10.
Example

t=3
delete(T,7)
Example

$t = 3$
delete(T, 7)

7 < 14 so the next node is the 1st child of the current node. This child has 3 > $t - 1$ elements, so no modification is needed.
Example

11 is the 2nd item of the node. Both the 2nd and 3rd children of the node have $2 = t - 1$ elements. Therefore we merge these children and move 7 to the new node.
Example

t=3
delete(T,7)
The element 7 is deleted from the current node.
Example

t=3
delete(T,4)
Example

4 < 14 so the next node is the 1st child of the current node. This child has 2 > \( t - 1 \) elements, so a modification is needed. This child has no immediate sibling with more than 2 = \( t - 1 \) elements, so perform merging on this child and its sibling. The height of the tree decreases by 1.
Example

$t=3$
delete(T,4)

3 < 4 < 10 so the next node is the 2nd child of the current node.
This child has $4 > t - 1$ elements, so no modification is needed.
Example

t=3
delete(T,4)
Example

t=3

delete(T,4)

The element 4 is deleted from the current node.
Example

t=3
delete(T,2)
Example

t=3
delete(T,2)

2 < 3 so the next node is the 1st child of the current node. This child has \(2 = t - 1\) elements, so a modification is needed. The immediate sibling of this child has \(3 > t - 1\) elements, so we perform shifting.
Example

t=3
delete(T,2)
Example

t=3
delete(T,2)

The element 2 is deleted from the current node.