# 作业反馈3-4

TC练习**18.1.1**；18.1.4；**18.2.3**；**18.2.4**；18.3.1

***18.1-1***

Why don't we allow a minimum degree of $t = 1$?

## *18.2-3*

Explain how to find the minimum key stored in a B-tree and how to find the predecessor of a given key stored in a B-tree.

- To find the minimum key in a B-tree
  - start at the root.
  - If the current node has children, then go to the leftmost child.
  - If the current node is a leaf, return the minimum key stored in the current node.

To find the predecessor of a given key, find the position of the given key in the tree and start at that node.

1. If the node is not a leaf, return the maximum key in the left subtree of the two subtrees that the key seperates.
2. If the node is a leaf and the key is not the minimum key in the leaf, return the maximum key in the node that is less than the given key.
3. If the node is a leaf and the key is the minimum key in the leaf, go to the father of the node. If in the father node there is a key that is less than the given key, return it. Otherwise go to the father of this node and repeat this process.
4. If in the above process we reach the root and still cannot find the predecessor, then the given key is the minimum key in the tree.

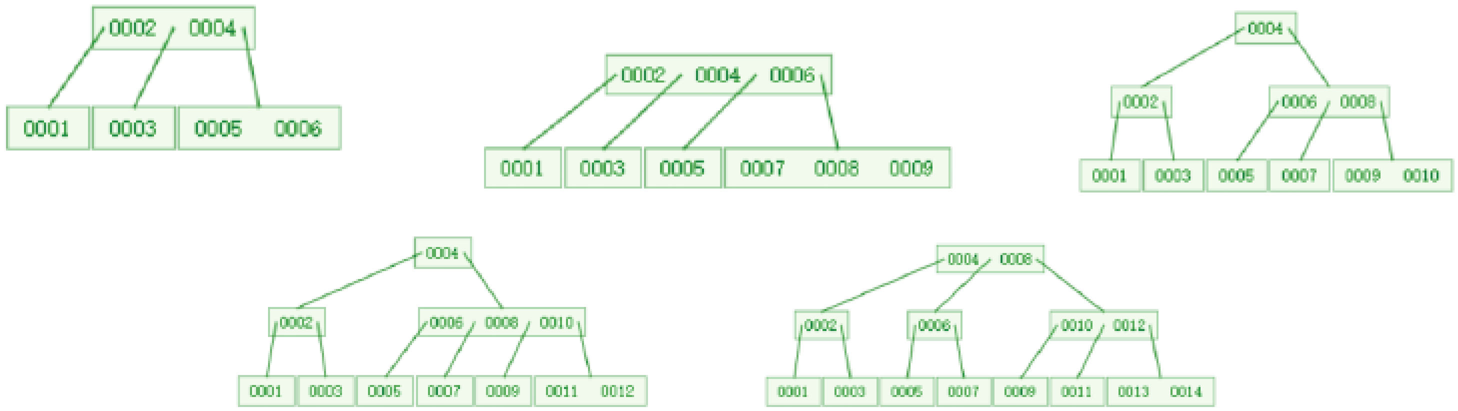
## *18.2-4* ★

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ into an empty B-tree with minimum degree 2. How many nodes does the final B-tree have?

- $\Theta(n)$?
  - 对，但是我不心甘！
  - 能否更加精确？

仅考虑 1-n 顺序插入的情况

由于 2-3-4 树的性质，在分裂时必定选择第 2 个 *key* 作为新的根节点并向上合并，这使得新节点的左边的部分只有一个 key，如图




并非书上介绍的single pass

直观上的说，如果插入是顺序的，那么只会改变整棵树右下角的节点，这使得整个树仅在最右支存在具有多个key的节点，并且根据分裂的性质，在最右支上，除了根节点可以有1 或2 或3个key，其他节点均有2 或3 个key

## 18.2-4 ★

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ into an empty B-tree with minimum degree 2. How many nodes does the final B-tree have?

Each node can have 1, 2, or 3 keys. Define the number of nodes that have at least 2 keys and 3 keys as $a$ and $b$, respectively. Then the number of nodes is given by $n - a - b$.

Only the nodes on the rightmost path can have at least 2 keys, so $a = h+1$ when the root has at least 2 keys and $a = h$ when the root has only 1 key, where $h$ is the height of the tree.

- $f(h)$ as the number of keys when the nodes on the rightmost path except the root have two keys and other nodes have only one key

$$f(h) = (2^h - 1) + 2(h+1) - 1 + 2(2^h - 1 - h)$$

$$= 3 * 2^h - 2$$

- $g(h)$ as the number of keys when all the nodes on the rightmost path have three keys and other nodes have one key.

$$g(h) = 3(2^h - 1) + 3(h+1) + 3(2^h - 1 - h)$$

$$= 6 * 2^h - 3$$

$g(h) = 3(2^h - 1) + 3 + g(h)$, when $h > 0$.

$g(0) = 3$.

## *18.2-4* ★

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ into an empty B-tree with minimum degree 2. How many nodes does the final B-tree have?

Since

$$f(h) \leq n \leq g(h)\,,$$

we have

$$\log_2 \frac{n+3}{6} \leq h \leq \log_2 \frac{n+2}{3}\,.$$

For any integer $n$, the following inequality holds,

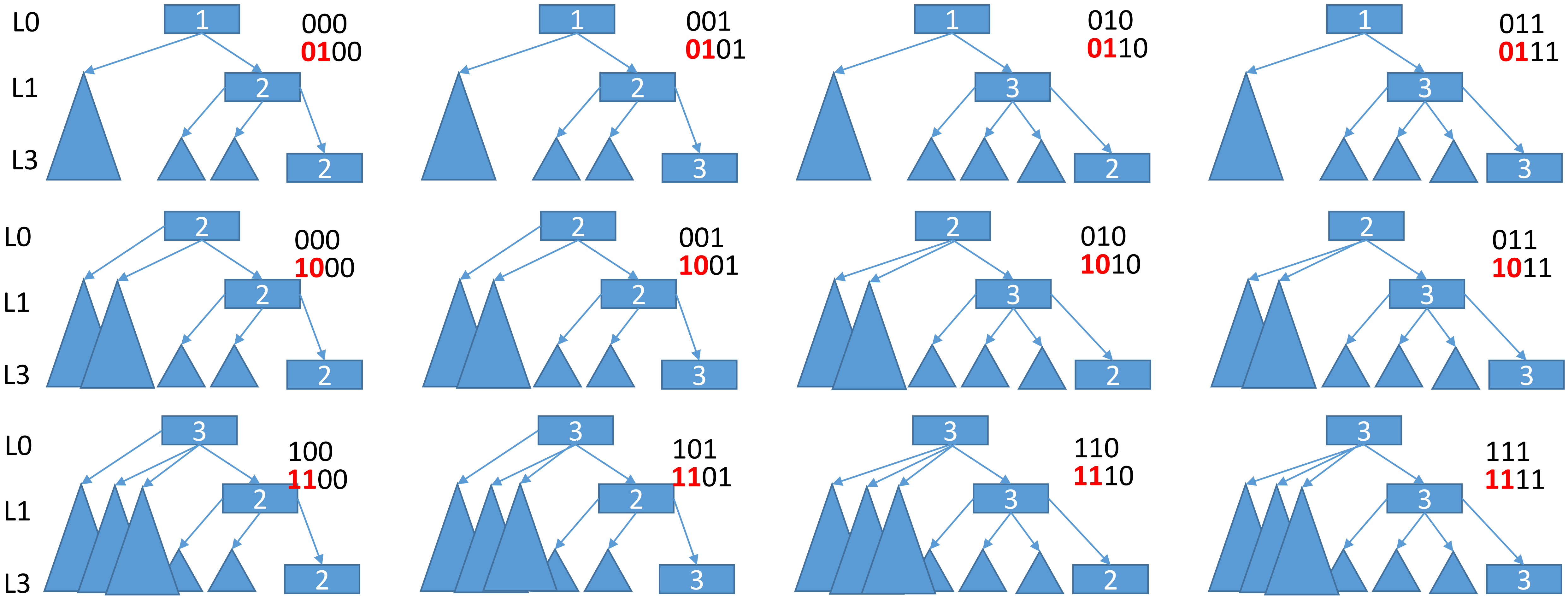
$$\log_2 \frac{n+2}{3} - \log_2 \frac{n+3}{6} < 1$$

Thus for any given integer $n$, there is only one integer $h$ that satisfies the above inequality, which is $\lfloor \log_2 \frac{n+2}{3} \rfloor$.

## *18.2-4* ★

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ into an empty B-tree with minimum degree 2. How many nodes does the final B-tree have?

Notice that when $n = f(h)$, there are no nodes that have 3 keys. If one more key $n + 1$ is inserted, the rightmost leaf of the tree will have 3 keys. If n + 2 is inserted, then the rightmost leaf will split and the father of it will have 3 keys.



## *18.2-4* ★

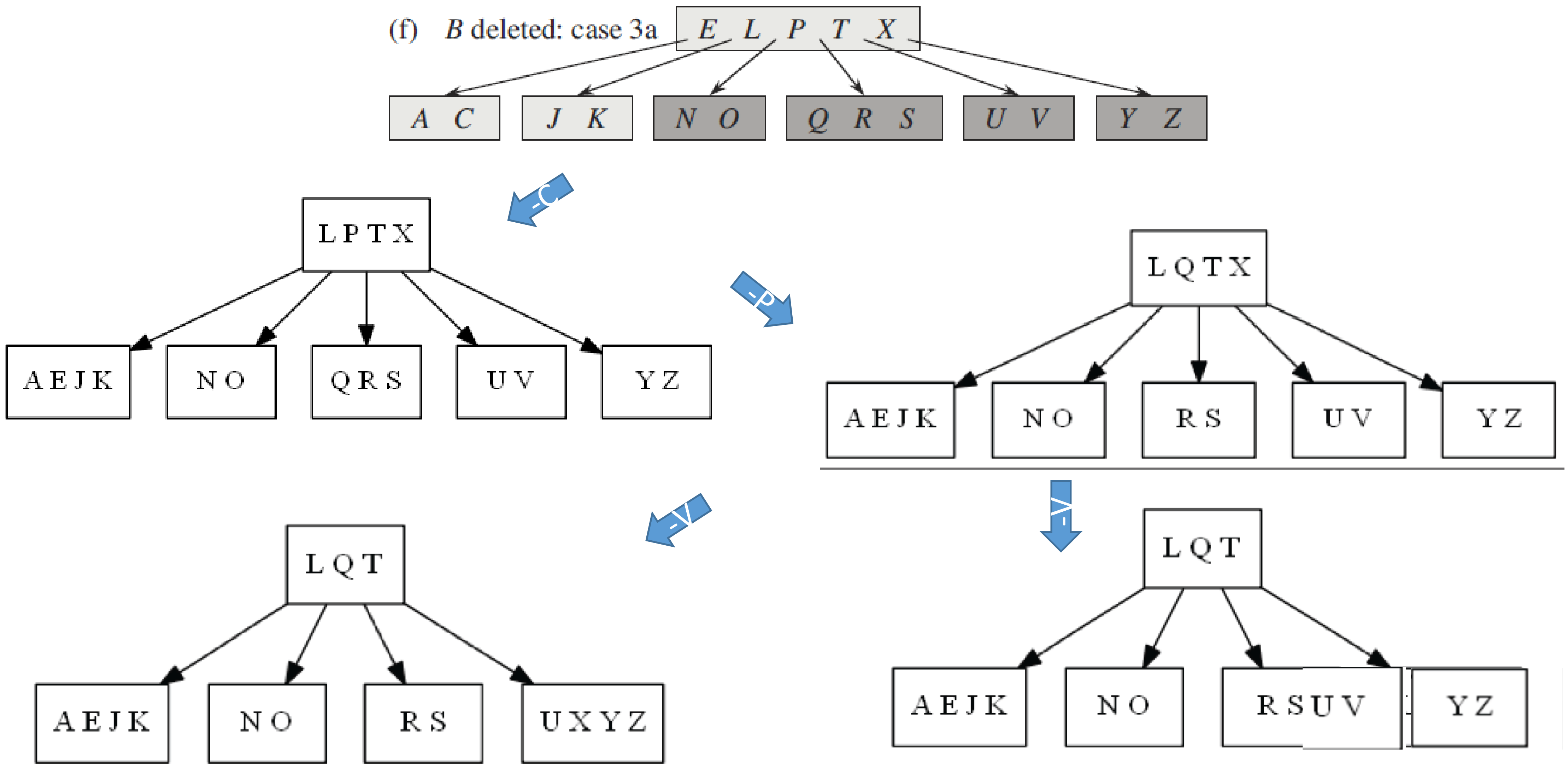
Suppose that we insert the keys $\{1, 2, \ldots, n\}$ into an empty B-tree with minimum degree 2. How many nodes does the final B-tree have?

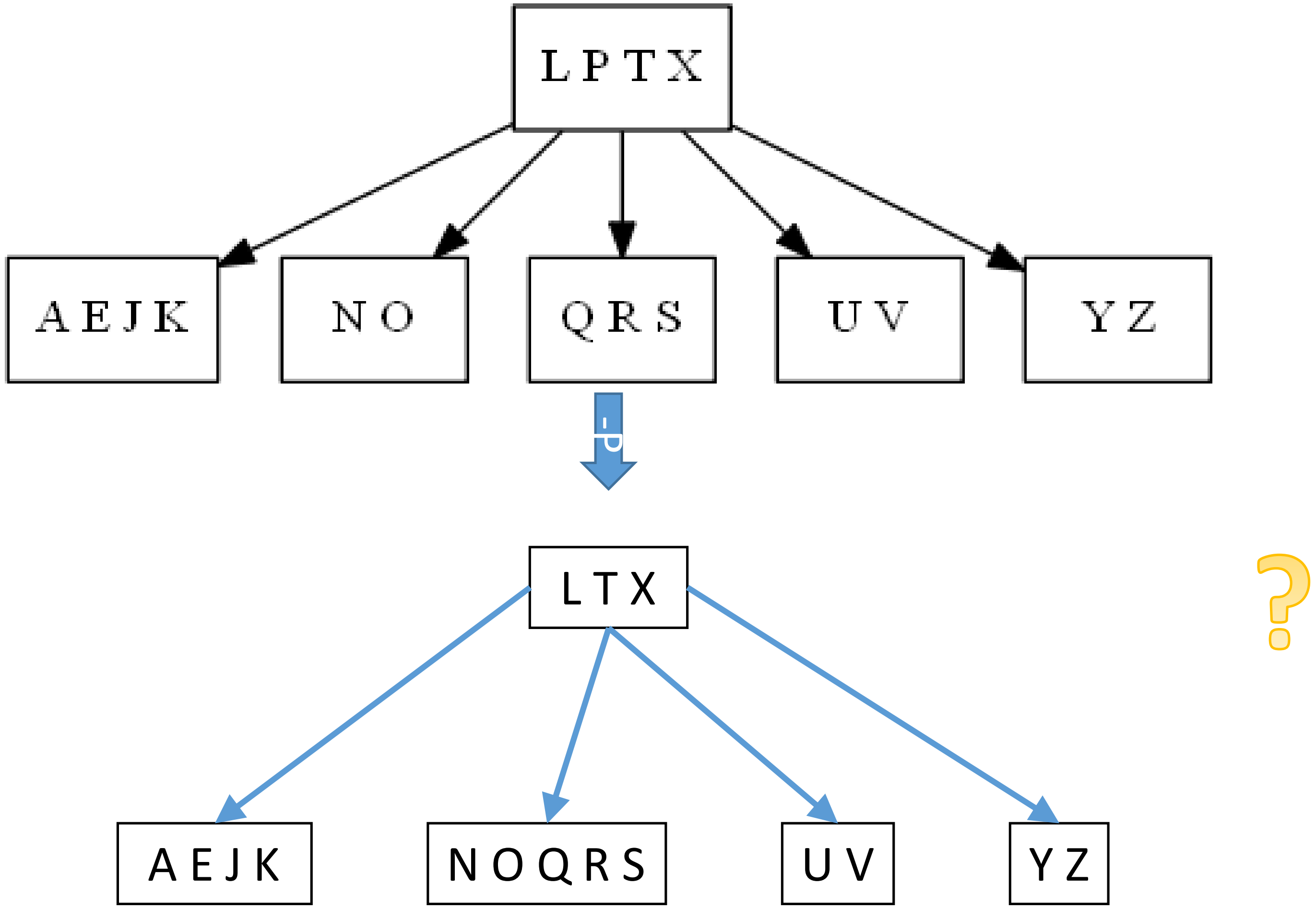
- 对于按序插入1~n，高度为h的满足条件的B-tree
  - 我们可以将$n - f(h)$表示为一个h+2位的二进制串；
  - 则包含3个key的节点个数b可以表示为：
    - $b = \left(value_{1\sim2}\left(n - f(h)\right) >= 3 ? 1 : 0\right) + bitcount_{3\sim h+2}\left(n - f(h)\right)$
    - 其中
      - $value_{1\sim2}\left(n - f(h)\right)$：$n - f(h)$的二进制表示(强制h+2位)前两位的值
      - $bitcount_{3\sim h+2}\left(n - f(h)\right)$：$n - f(h)$二进制表示(强制h+2位)的3~h+2位中1的个数

$$\#Node(n) = n - a - b$$

### *18.3-1*

Show the results of deleting $C$, $P$, and $V$, in order, from the tree of Figure 18.8(f).




LPTX
AEJK
NO
QRS
UV
YZ
-P
LTX
AEJK
NOQRS
UV
YZ