B+ Trees

What's wrong with sequential index?

+ Seq scan is fast.

– costly to maintain

B-Trees/B+Trees: B_?_?_Trees

Balanced

- Intuition:
  - Give up on sequentiality of index
  - Try to get "balance" by dynamic reorganization

B-tree 1972

- B+trees:
  - Textbook refers to B+trees (a popular variant) as B-trees (as most people do)
  - Distinction will be clear later (ok to confuse now)

B+-tree

Behind the Scene:
UIUC (Alumni) Contribution!

Prof. Rudolf Bayer

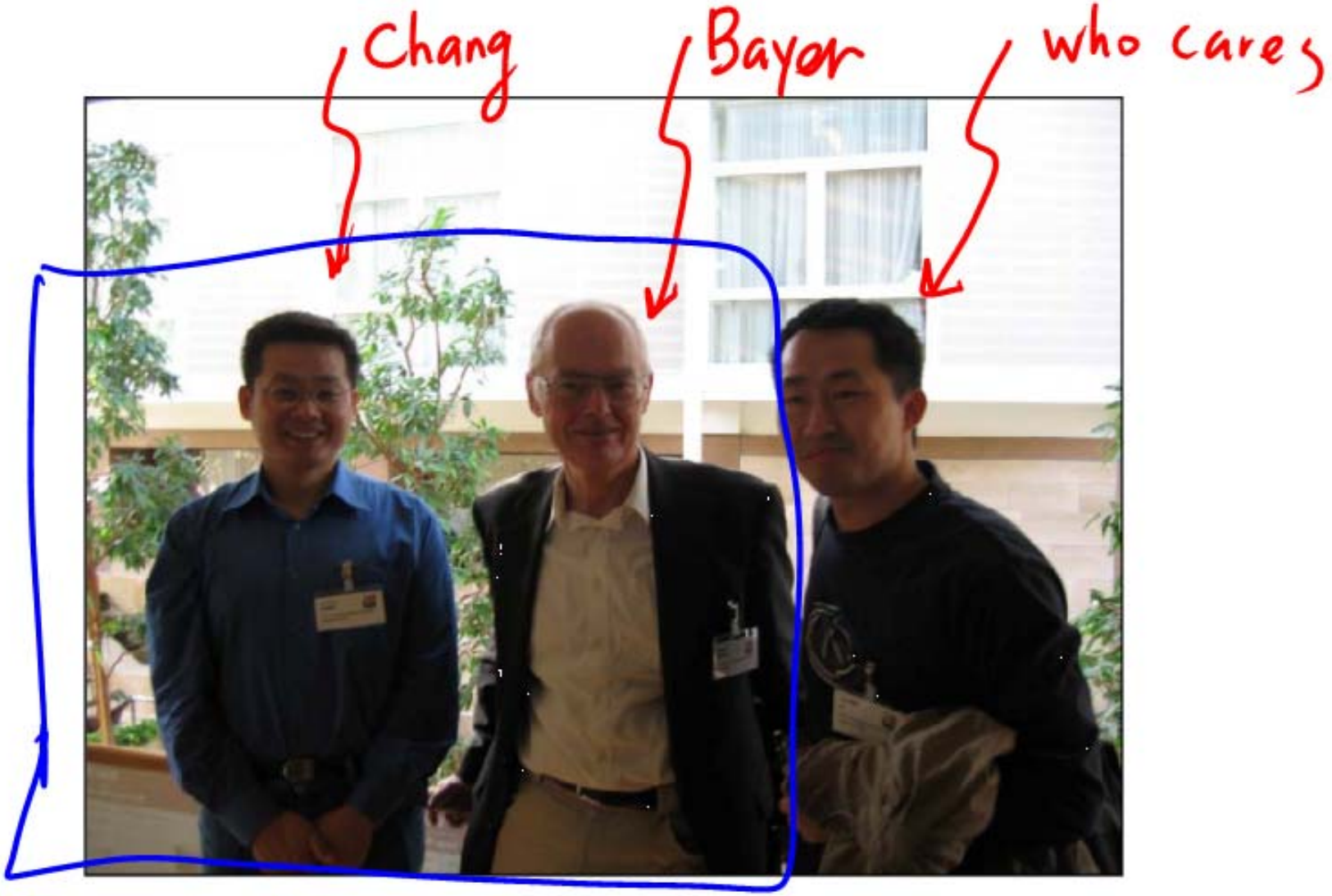
Rudolf Bayer studied Mathematics in Munich and at the University of Illinois, where he received his Ph.D. in 1966. After working at Boeing Research Labs he became an Associate Professor at Purdue University. He is a Professor of Informatics at the Technische Universität München since 1972 and … …

The 2001 SIGMOD Innovations Award goes to Prof. Rudolf Bayer of the Technical University of Munich, for his invention of the B-Tree (with Edward M. McCreight), of B-Tree prefix compression, and of lock coupling (a.k.a. crabbing) for concurrent access to B-Trees (with Mario Schkolnick). All of these techniques are widely used in commercial database products. ……

The Original Publication

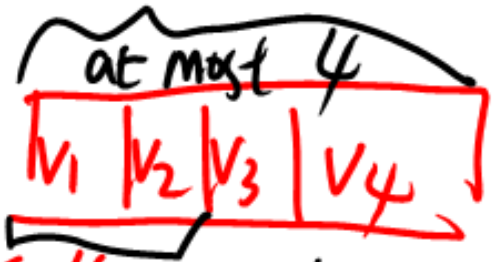
Rudolf Bayer, Edward M. McCreight: Organization and Maintenance of Large Ordered Indices. Acta Informatica 1: 173-189 (1972)

Behind the Scene: And he said Hello!



B+ Trees Basics

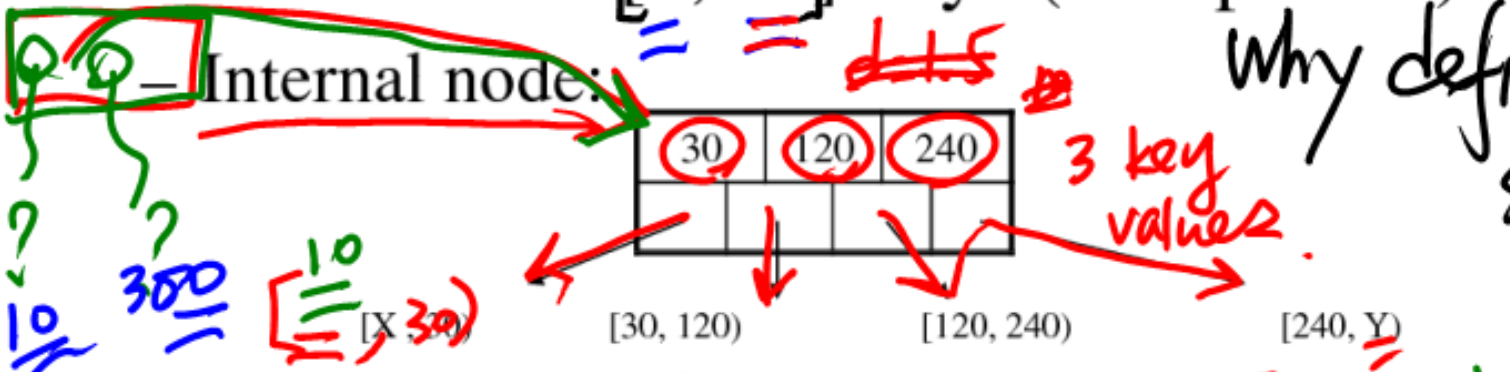
(B-tree) a.w. too empty



$n = 2d$

$d=2 \Rightarrow 2 \le v \le 4$ 2 values at least

- Parameter d = the *degree*
- Each node has [d, 2d] keys (except root)
  - Internal node:

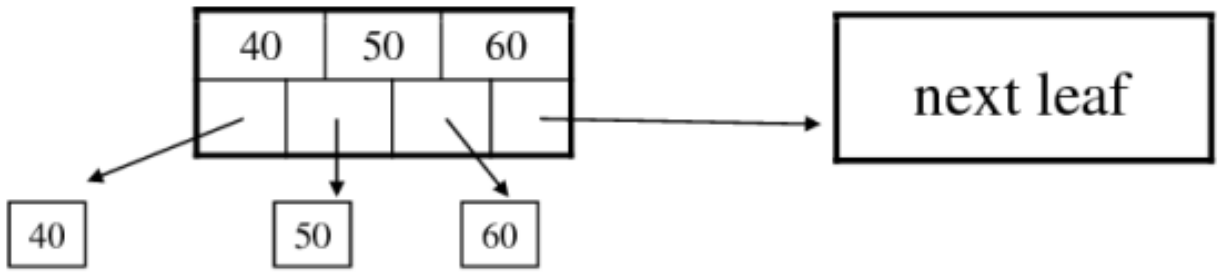


3 key values.

Why define capacity? Store index

⇒ size of each node = 1 "page" "block" (4k, 8k)

  - Leaf:



Student (name, age)

Bt Tree for age

B+ Tree Example

Select name dec. by
from stu
where age = 30

value

age=30

"pinned"

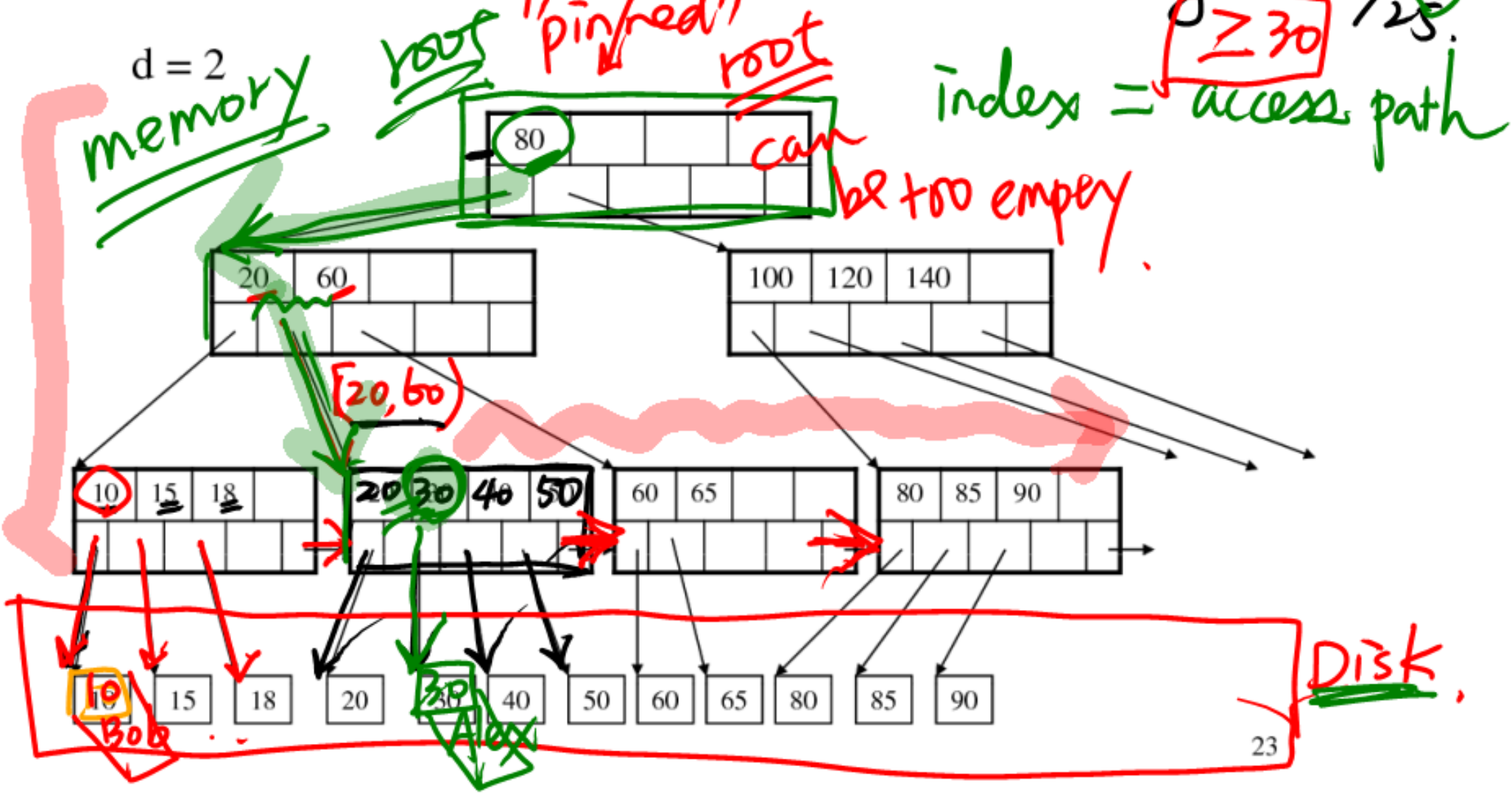
root

root can be too empty.

index = access path

d = 2

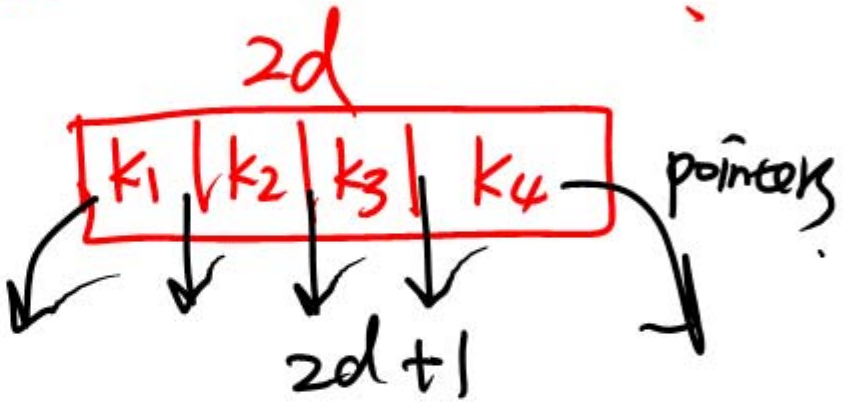
memory



B+ Tree Design

d, 2d max # of keys

- How large d ? ⇒ as large as u can fit on one disk block
- Example: (integer)
  - Key size = 4 bytes
  - Pointer size = 8 bytes ("address")
  - Block size = 4096 byes



- keys: 2d x 4 + (2d+1) x 8 <= 4096 (ptrs)
- d = 170 → at most 2d = 340 keys. 2d+1 = 341 ptrs. (blok size)

Searching a B+ Tree

- Exact key values:
  - Start at the root
  - Proceed down, to the leaf

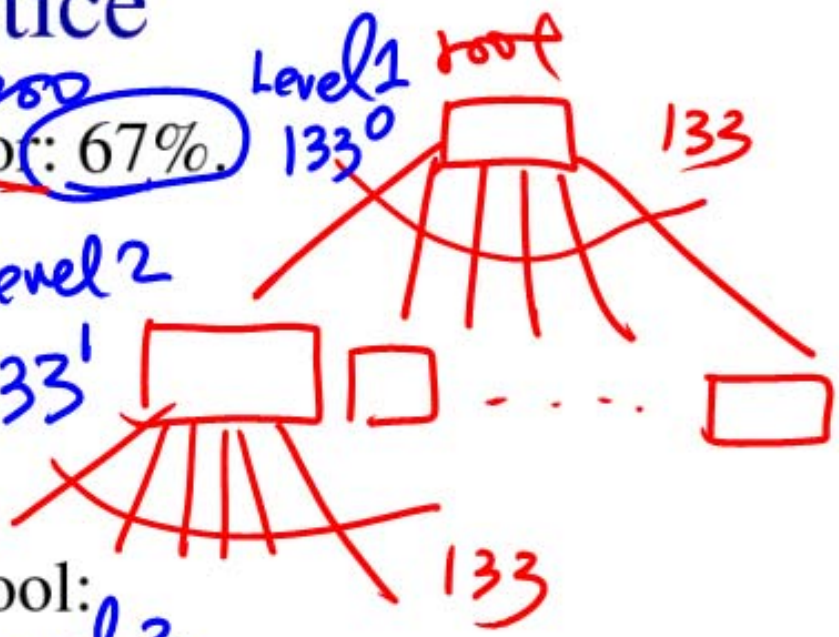
```
Select name
From people
Where age = 25
```

- Range queries:
  - As above
  - Then sequential traversal

```
Select name
From people
Where 20 <= age
  and  age <= 30
```

B+ Trees in Practice

- Typical order: 100. Typical fill-factor: 67%.
  - average fanout = 133
- Typical capacities:
  - Height 4: 133^4 = 312,900,700 records
  - Height 3: 133^3 = 2,352,637 records
- Can often hold top levels in buffer pool:
  - Level 1 = 1 page = 8 Kbytes
  - Level 2 = 133 pages = 1 Mbyte
  - Level 3 = 17,689 pages = 133 MBytes



d=100 100 ≤ ≤ 200

200×67%

Level 3 133^2

Level 4 $133^3 \geq 2$ million

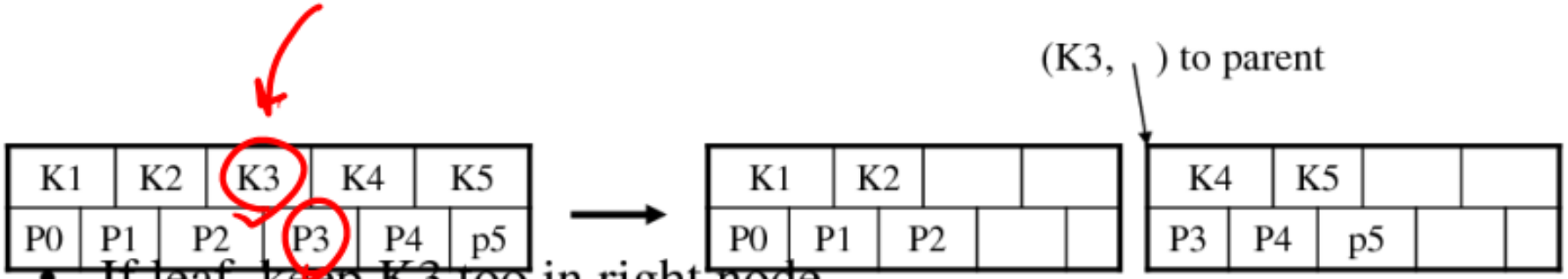
~ top level in mem

~ tree short

Insertion in a B+ Tree

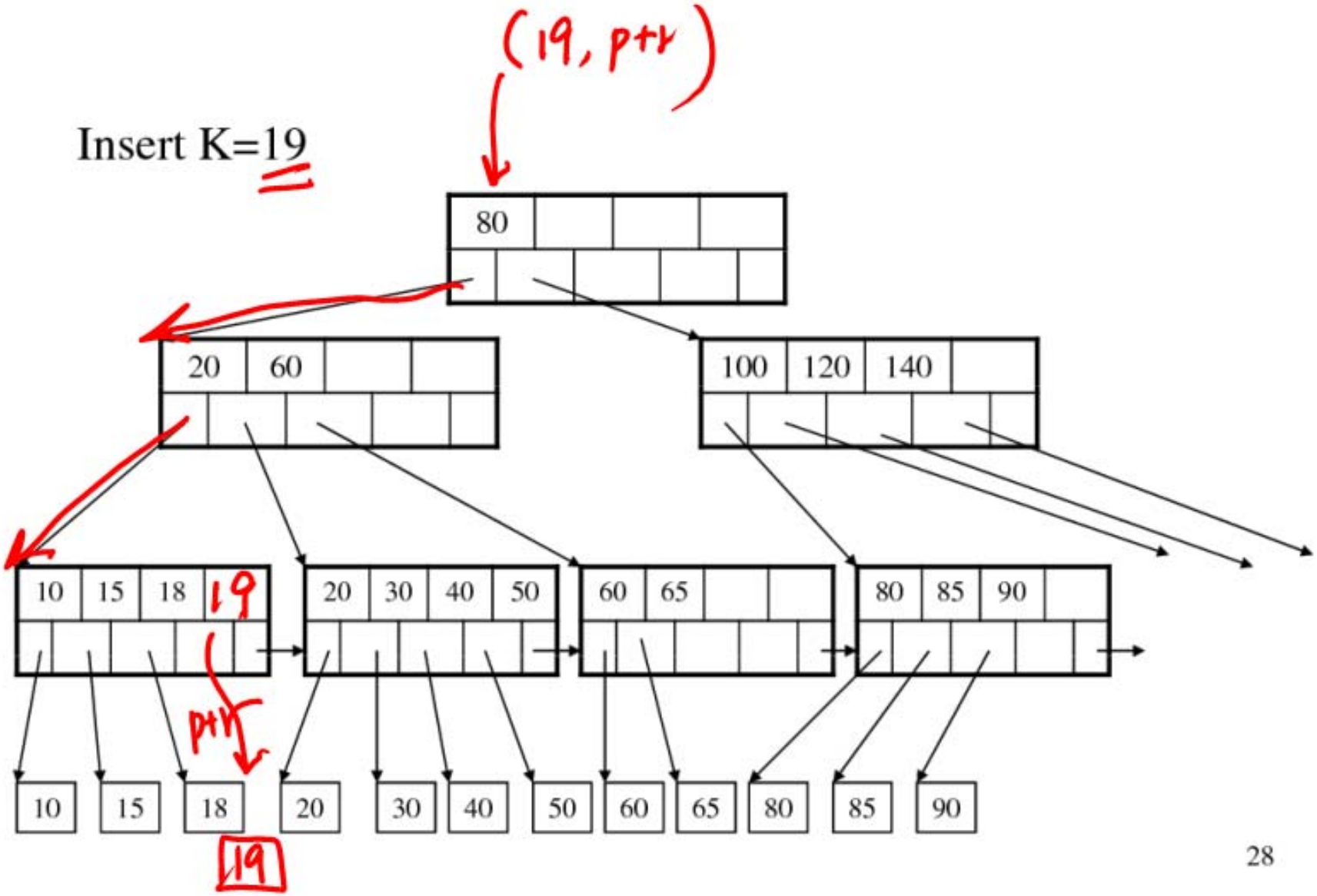
Insert (K, P)

- Find leaf where K belongs, insert
- If no overflow (2d keys or less), halt
- If overflow (2d+1 keys), split node, insert in parent:



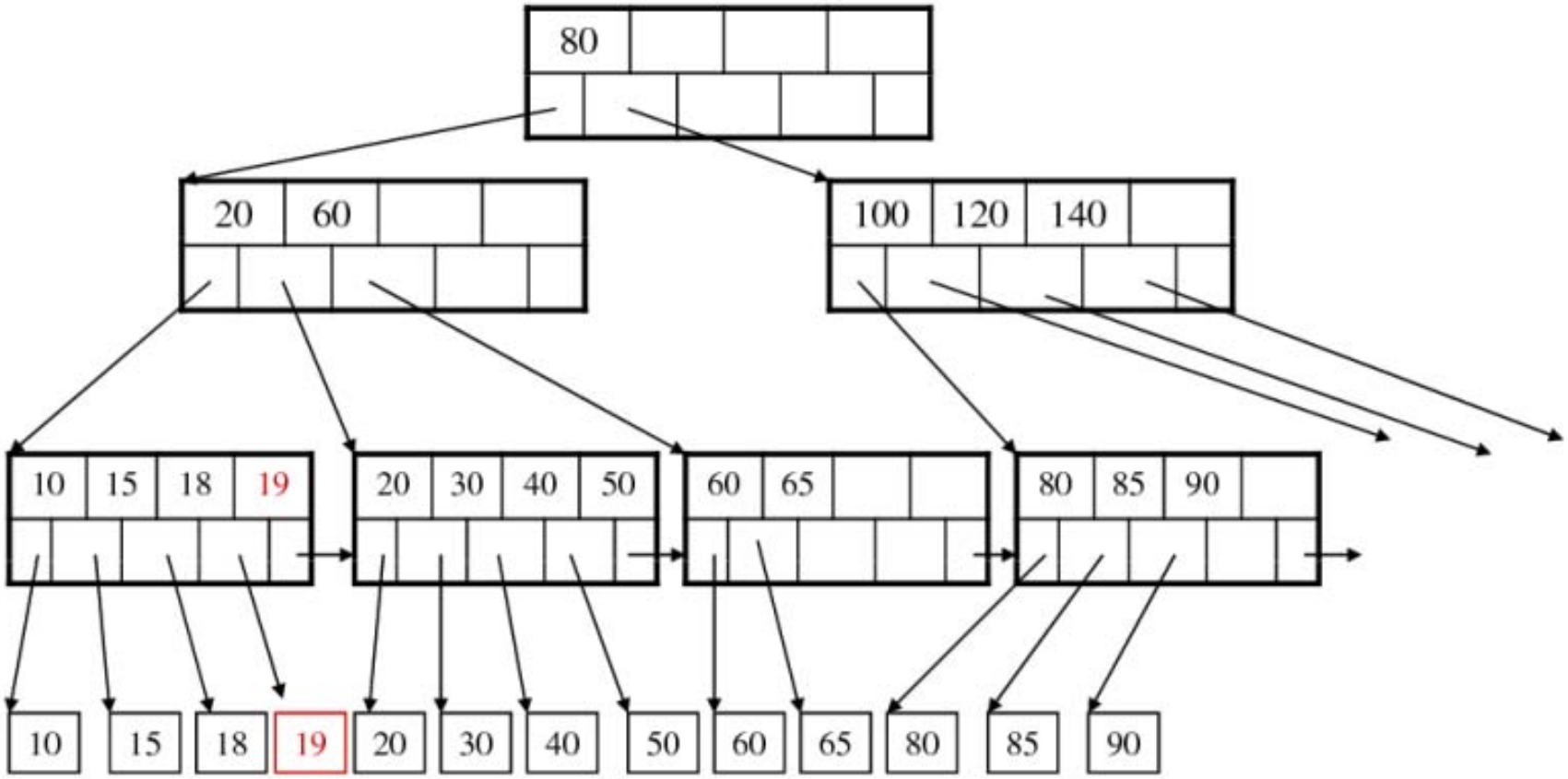
- If leaf, keep K3 too in right node
- When root splits, new root has 1 key only
  - that's why root is special for degree satisfaction

Insertion in a B+ Tree



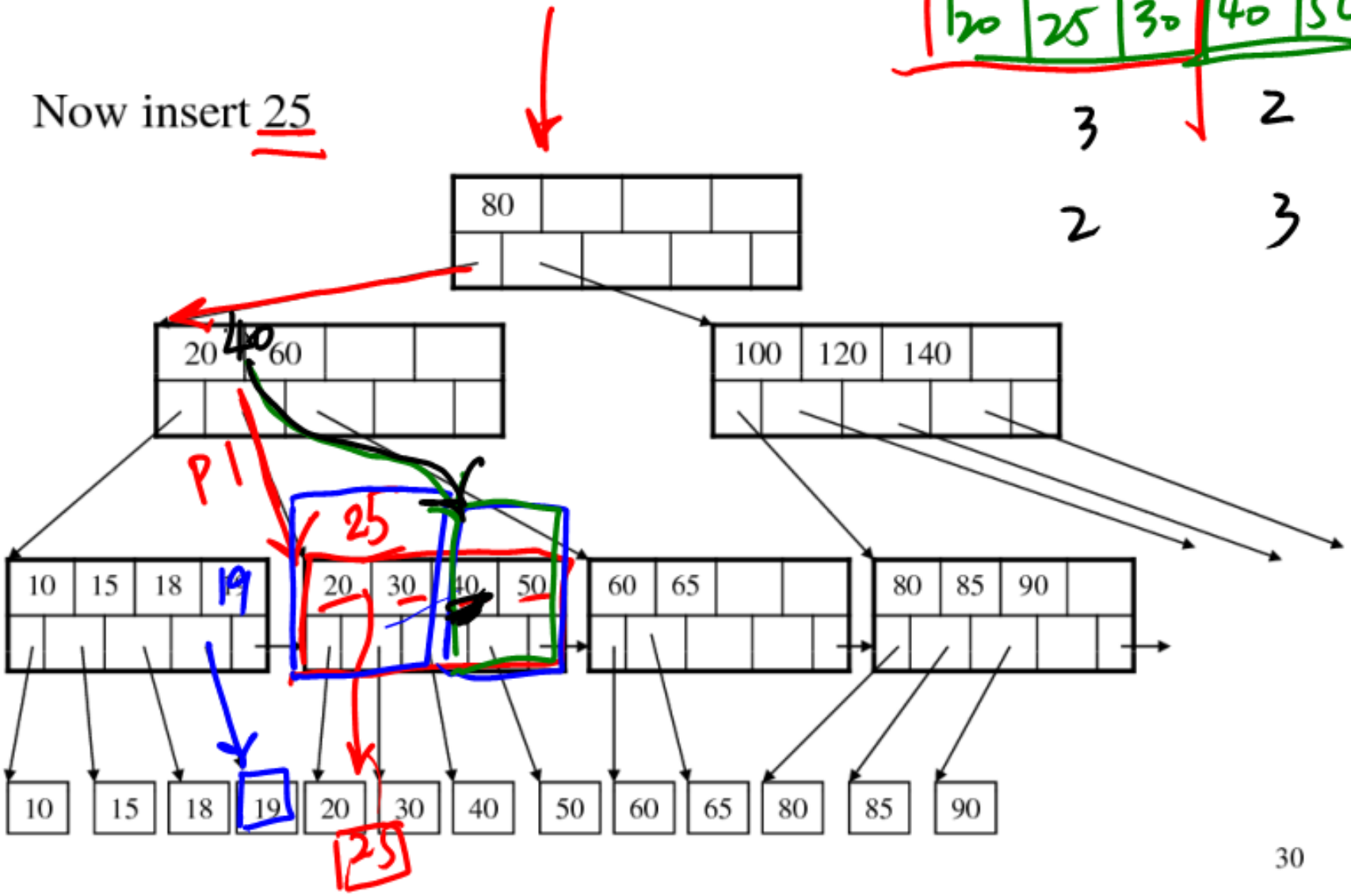
Insertion in a B+ Tree

After insertion



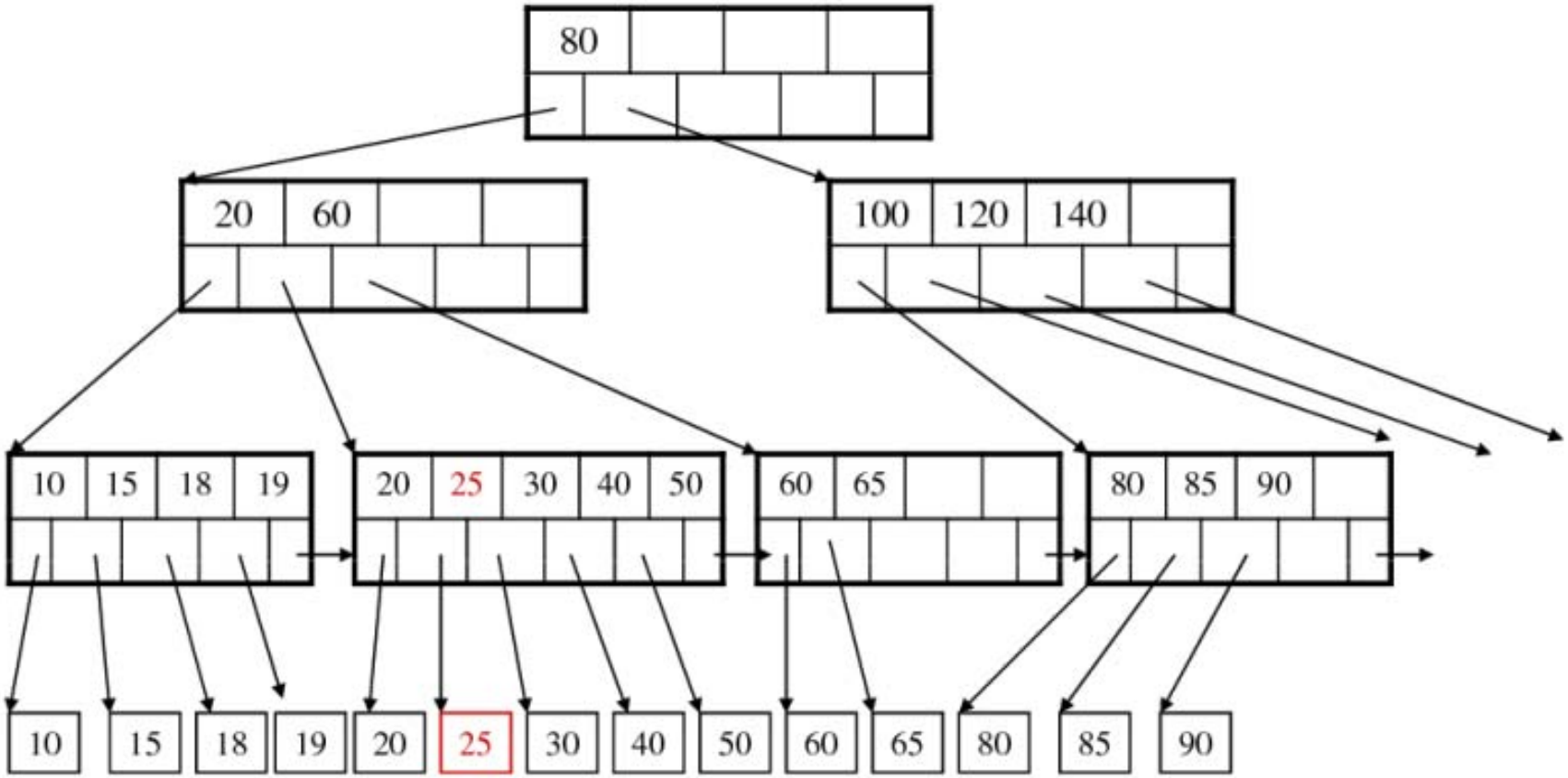
Insertion in a B+ Tree

Now insert 25



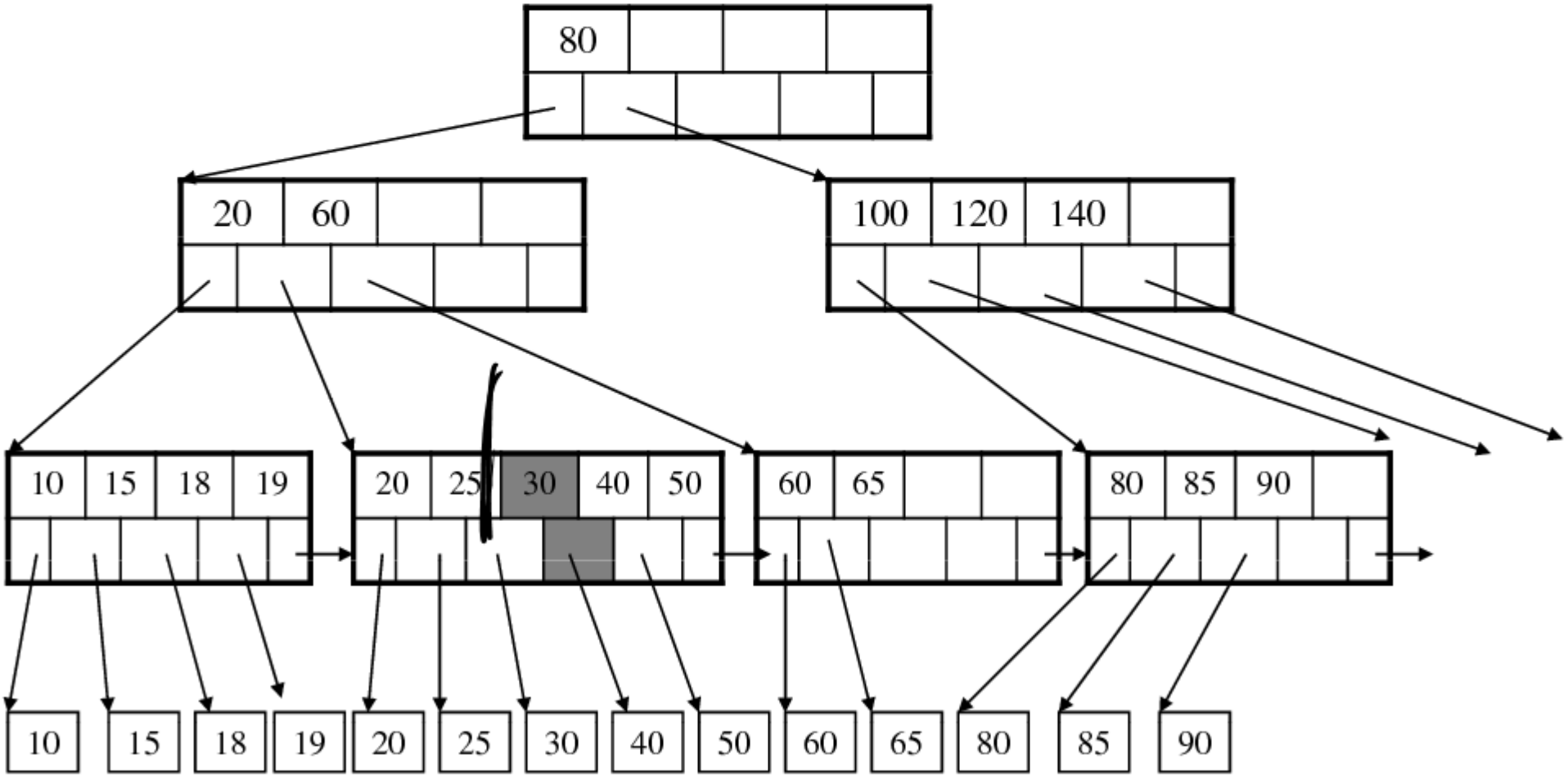
Insertion in a B+ Tree

After insertion



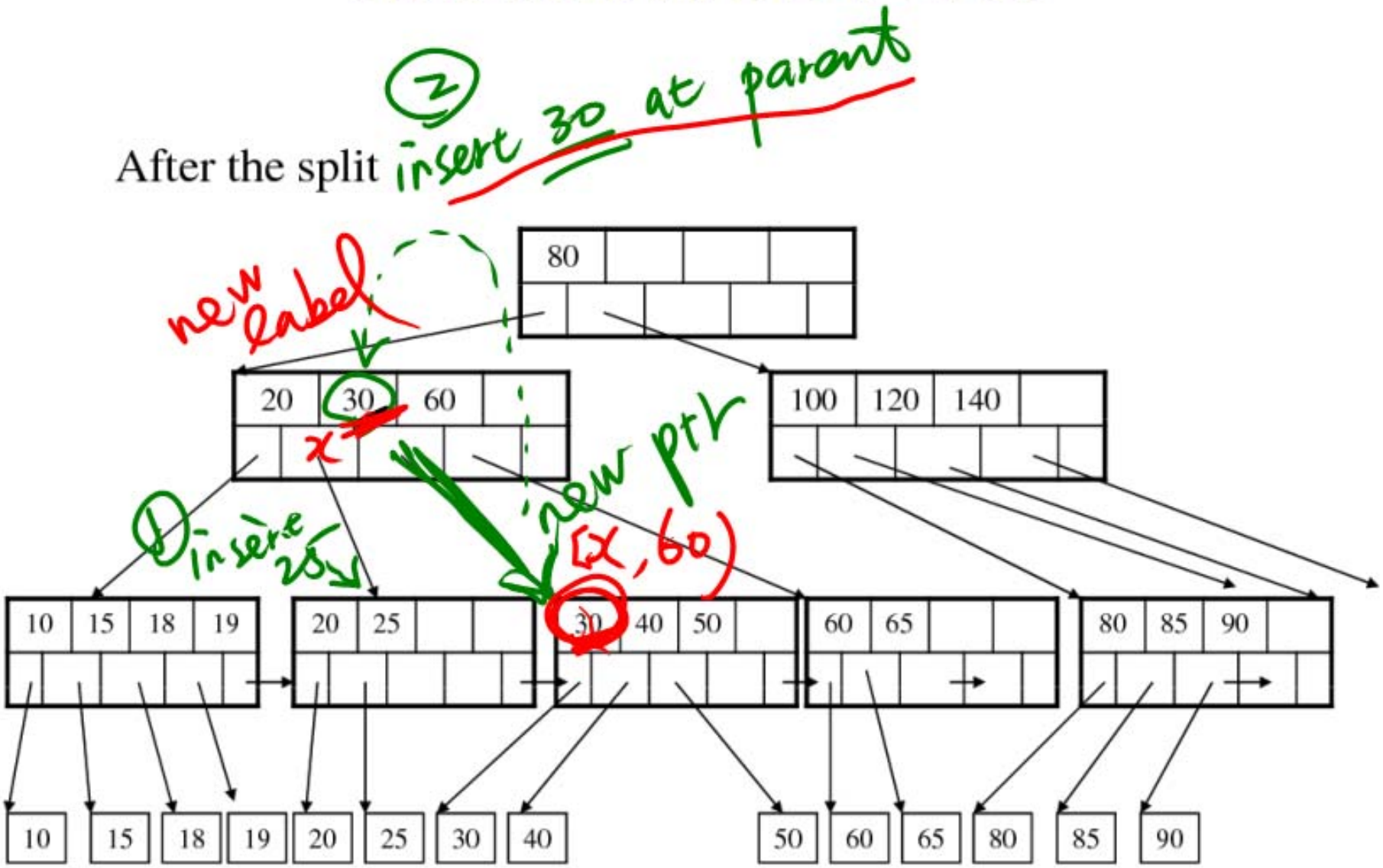
Insertion in a B+ Tree

But now have to split !



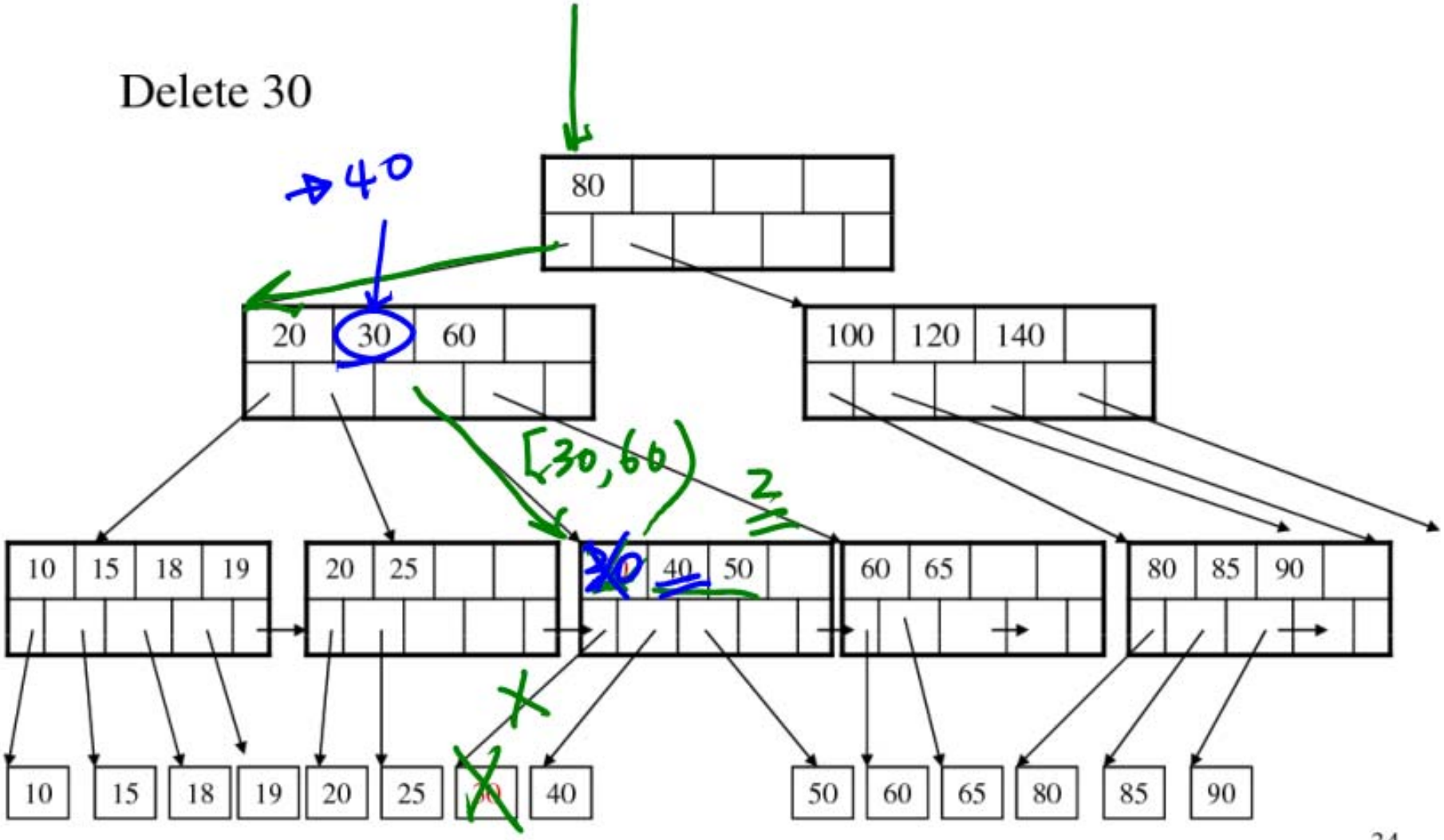
Insertion in a B+ Tree

After the split

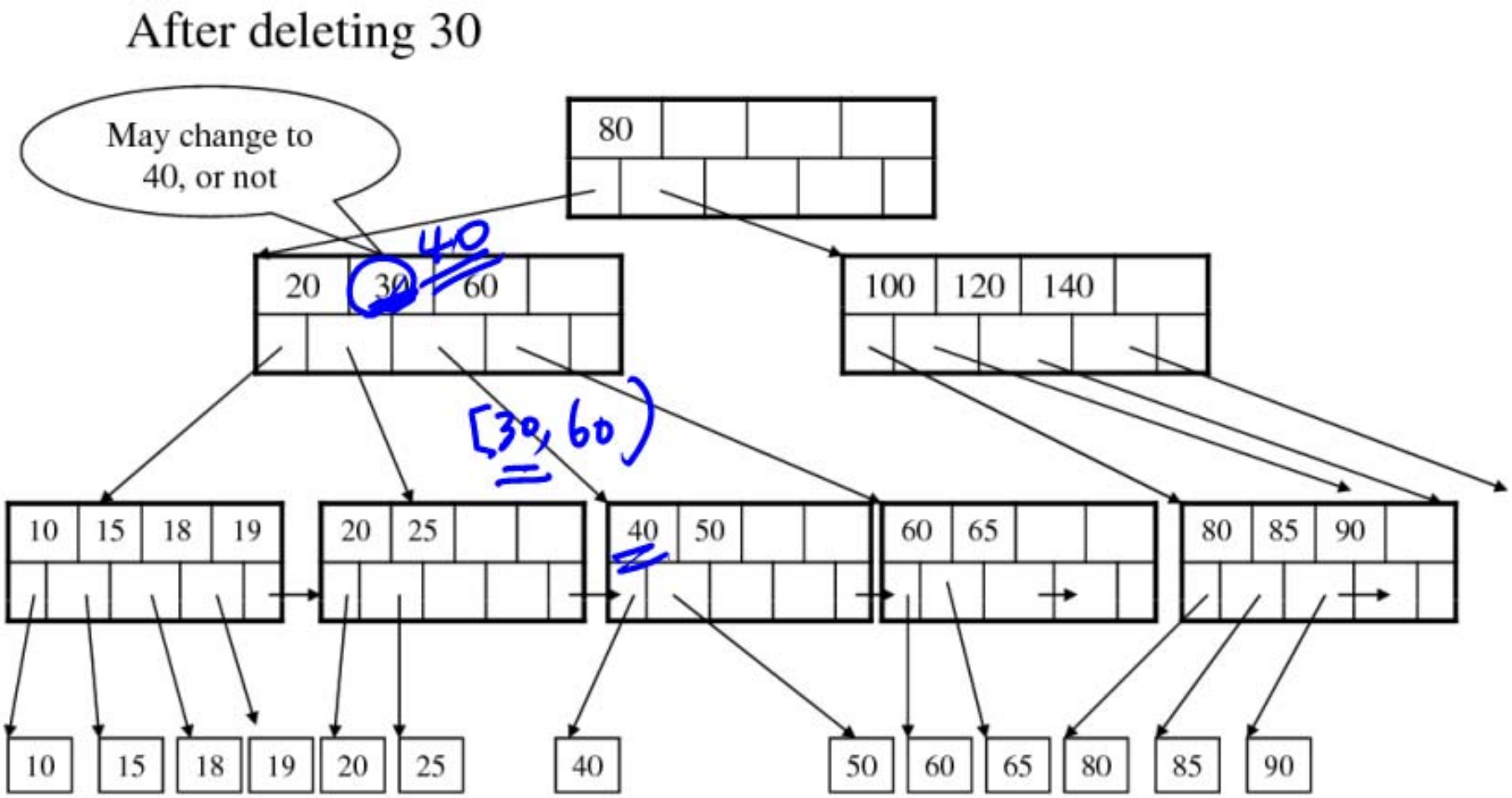


Deletion from a B+ Tree

Delete 30



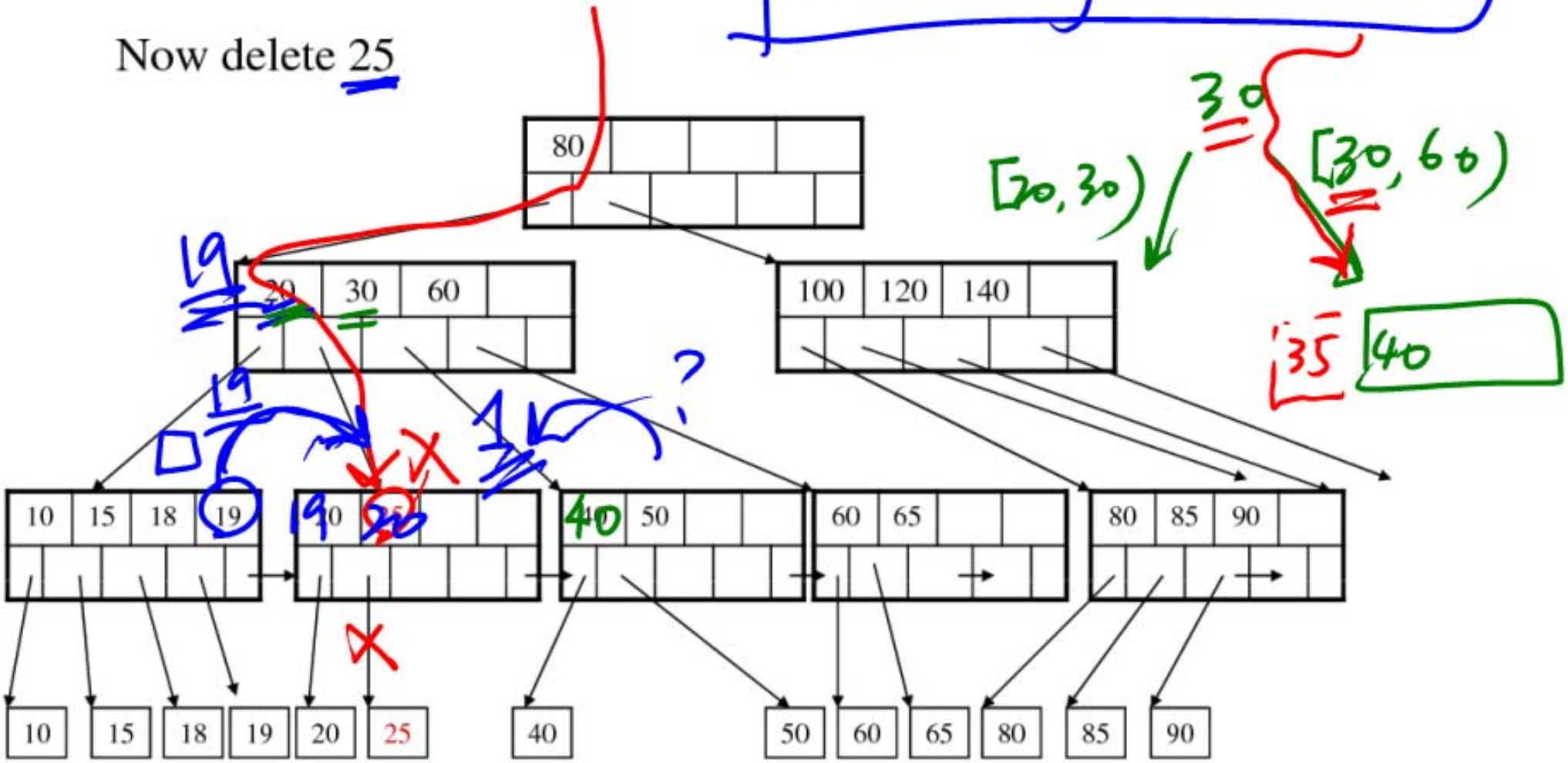
Deletion from a B+ Tree



Deletion from ~~a B+ Tree~~

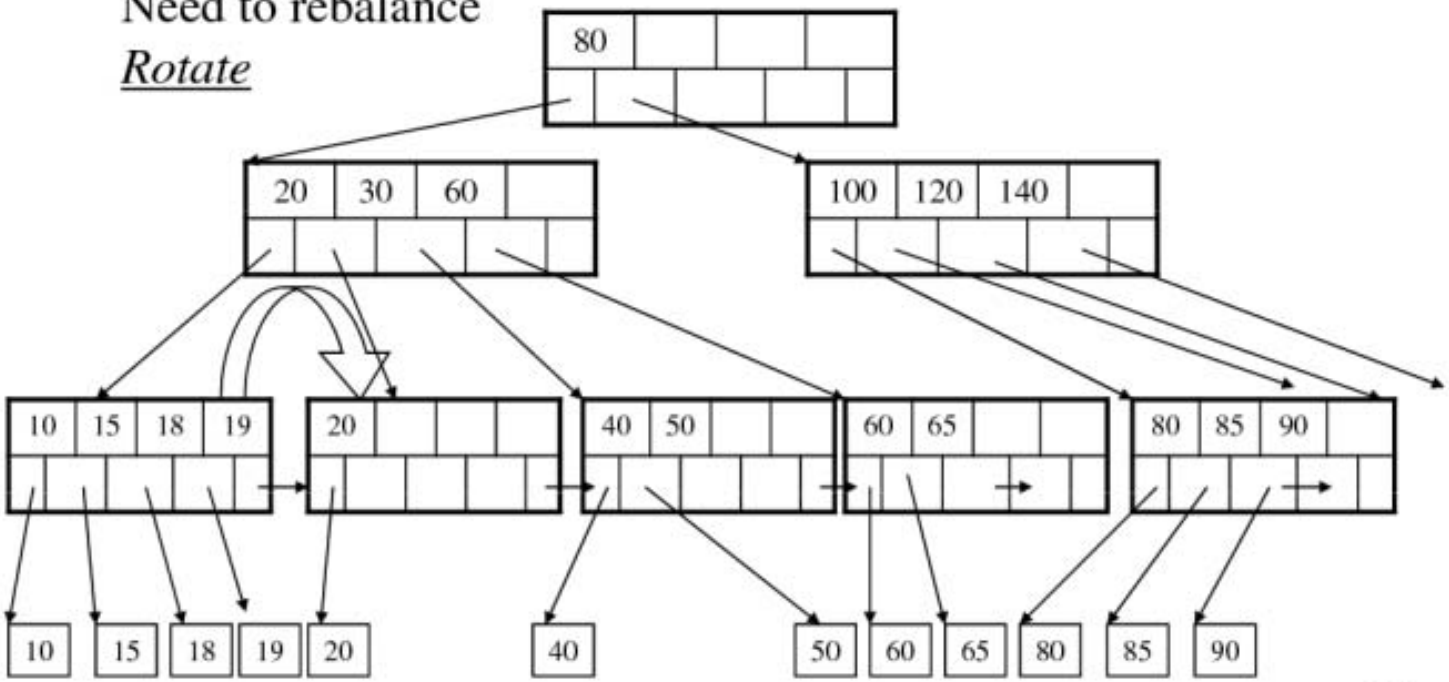
Rotate if u can

Now delete 25



Deletion from a B+ Tree

After deleting 25
Need to rebalance
Rotate

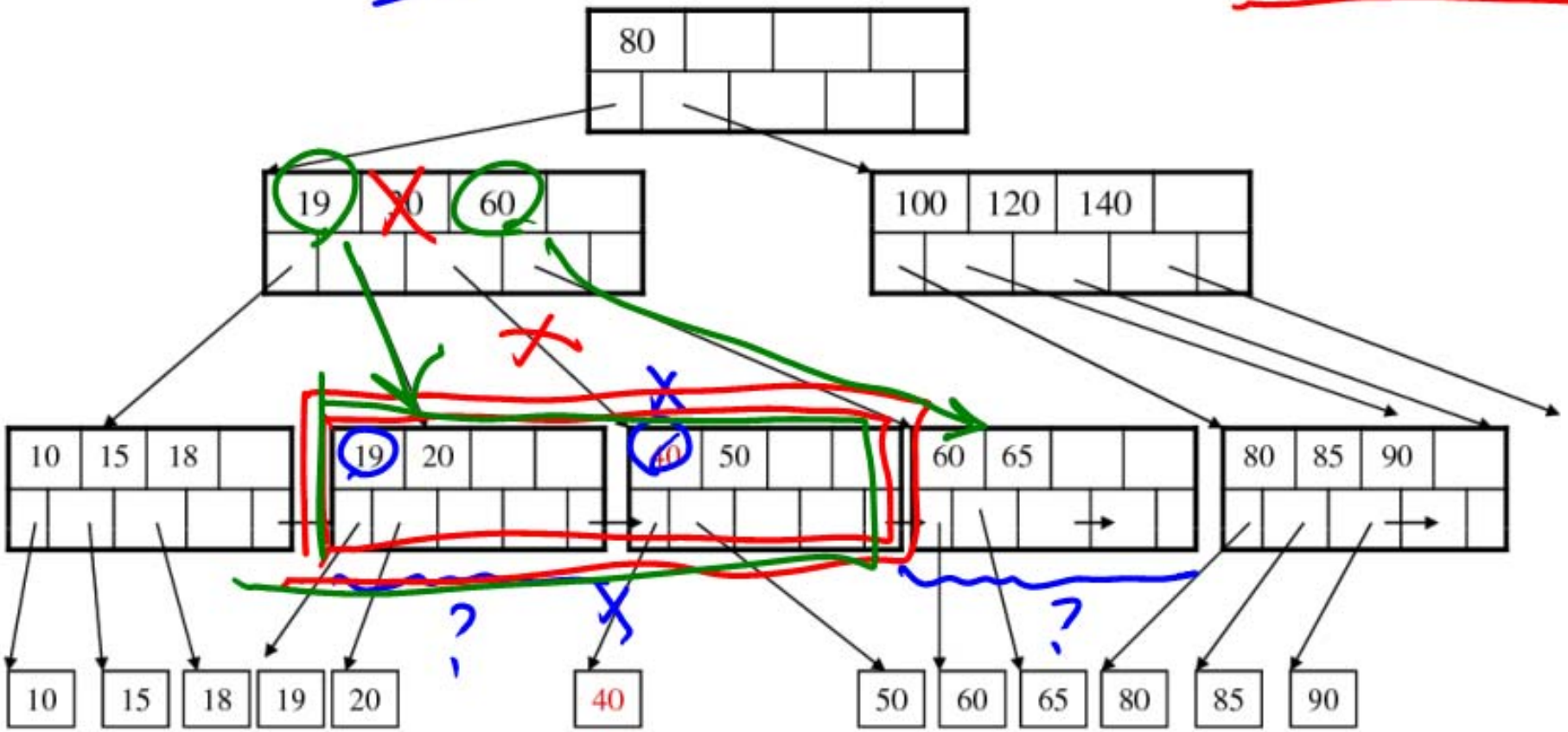


Deletion from a B+ Tree

Merge if you have to.

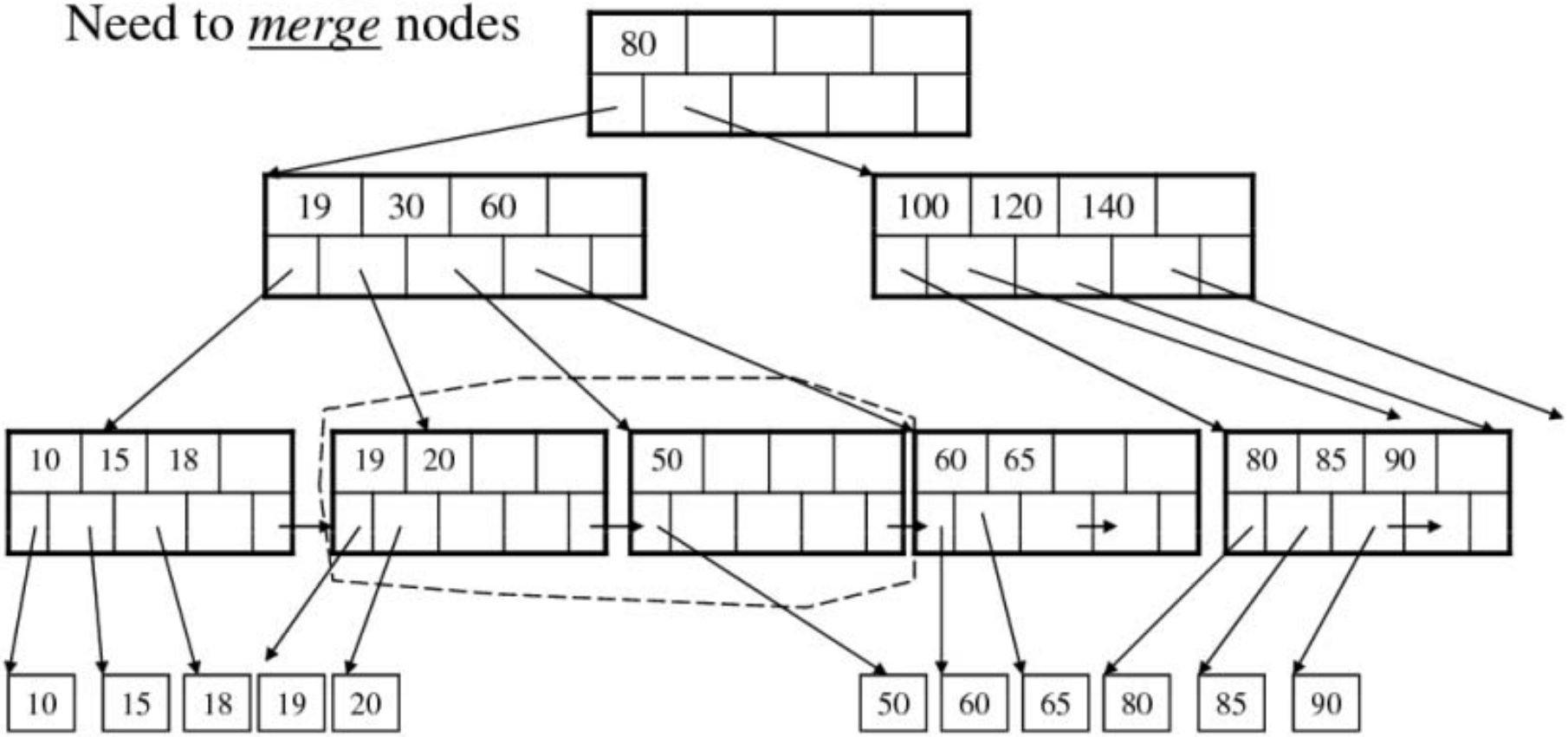
19	20	50

Now delete 40



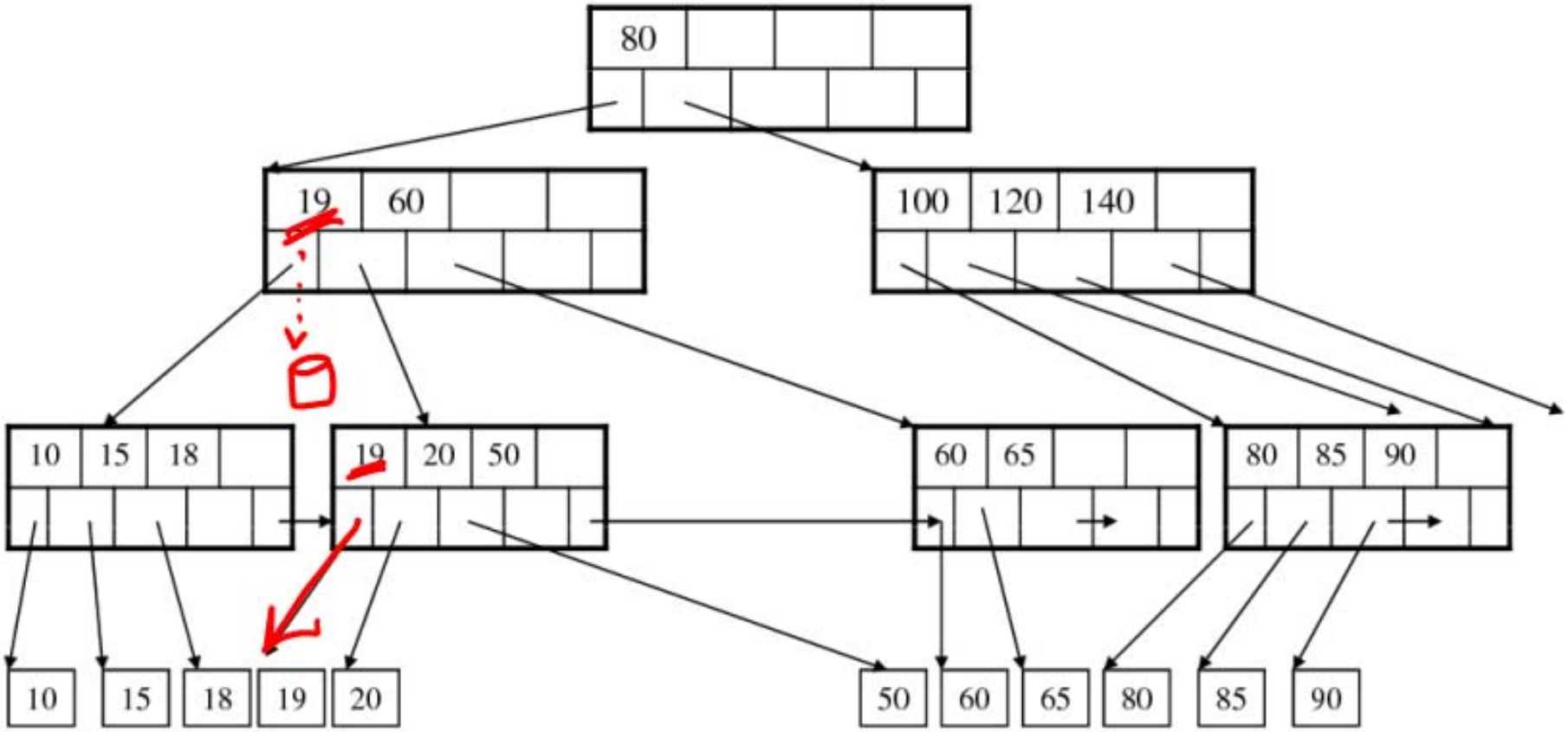
Deletion from a B+ Tree

After deleting 40
Rotation not possible
Need to *merge* nodes



Deletion from a B+ Tree

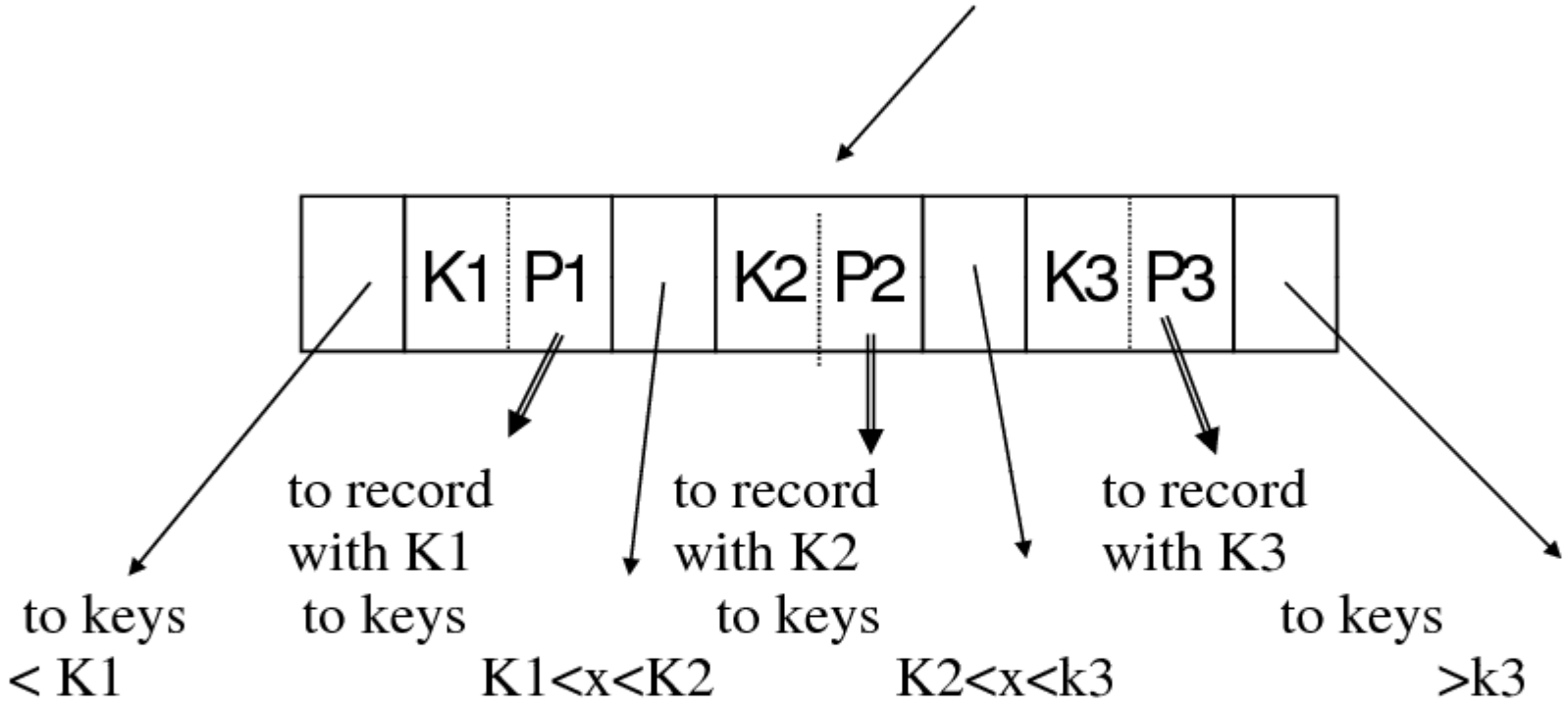
Final tree



Variation on B+tree: B-tree (no +)

- Idea:
  - Avoid duplicate keys
  - Have record pointers in non-leaf nodes

- Note: Textbook's B-Tree means B+-tree!

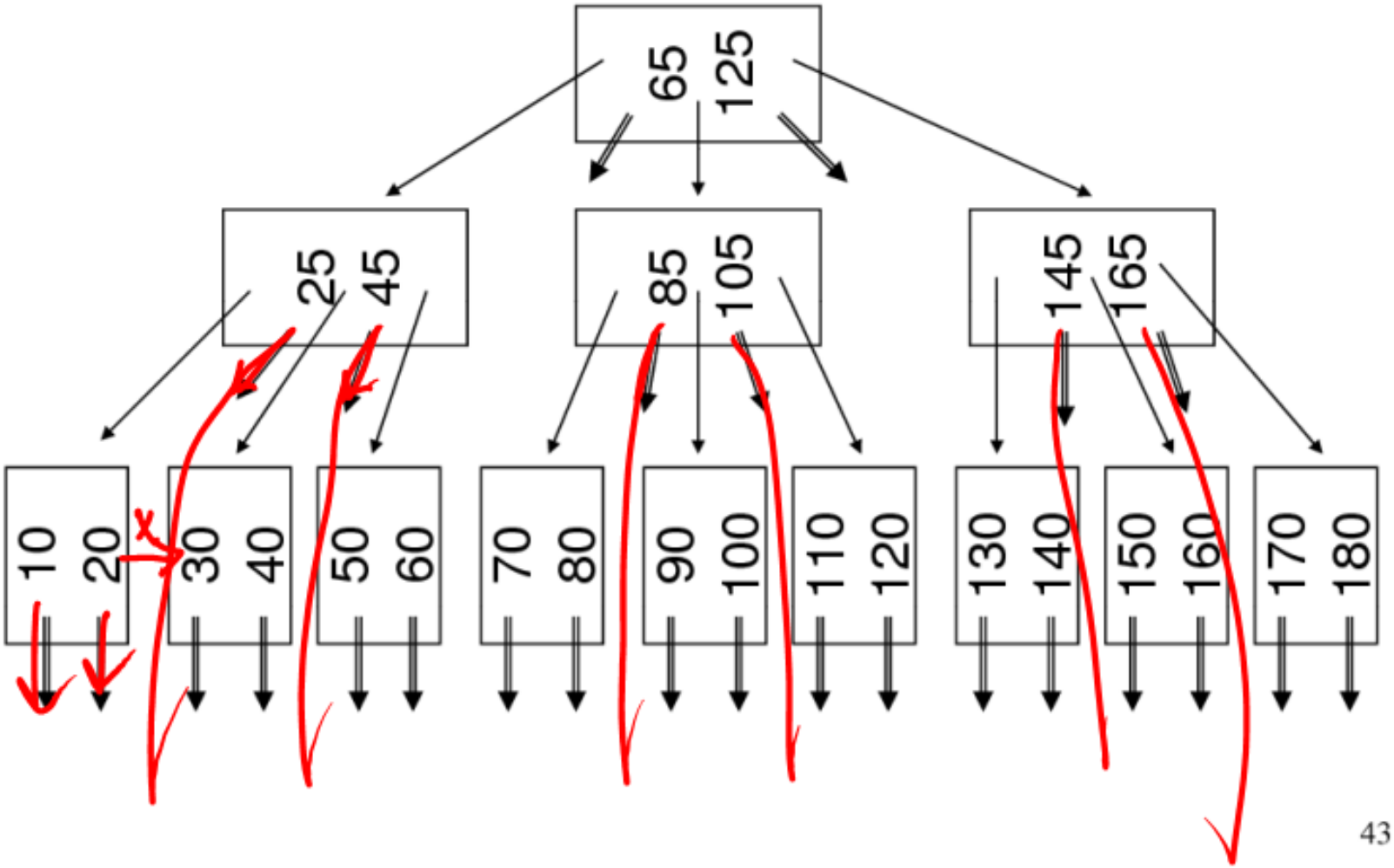

K1
P1
K2
P2
K3
P3
to record with K1
to record with K2
to record with K3
to keys < K1
to keys K1<x<K2
to keys K2<x<k3
to keys >k3

B-tree example

B-Tree (X B+-tree)

n=2

- Sequence pointers not useful now!



Quiz

Name: _____ NetID: _____

Q. How do u like special topics?

SQLite : 1 -3-- 5

dislike like

SQL Tuing 1 ---- 5

Q. Suggestions for Special Topics?

e.g. stop it.

do it, but ....