Data Structures and Algorithms (11)

Instructor: Ming Zhang
Textbook Authors: Ming Zhang, Tengjiao Wang and Haiyan Zhao
Higher Education Press, 2008.6 (the "Eleventh Five-Year" national planning textbook)

https://courses.edx.org/courses/PekingX/04830050x/2T2014/

Content

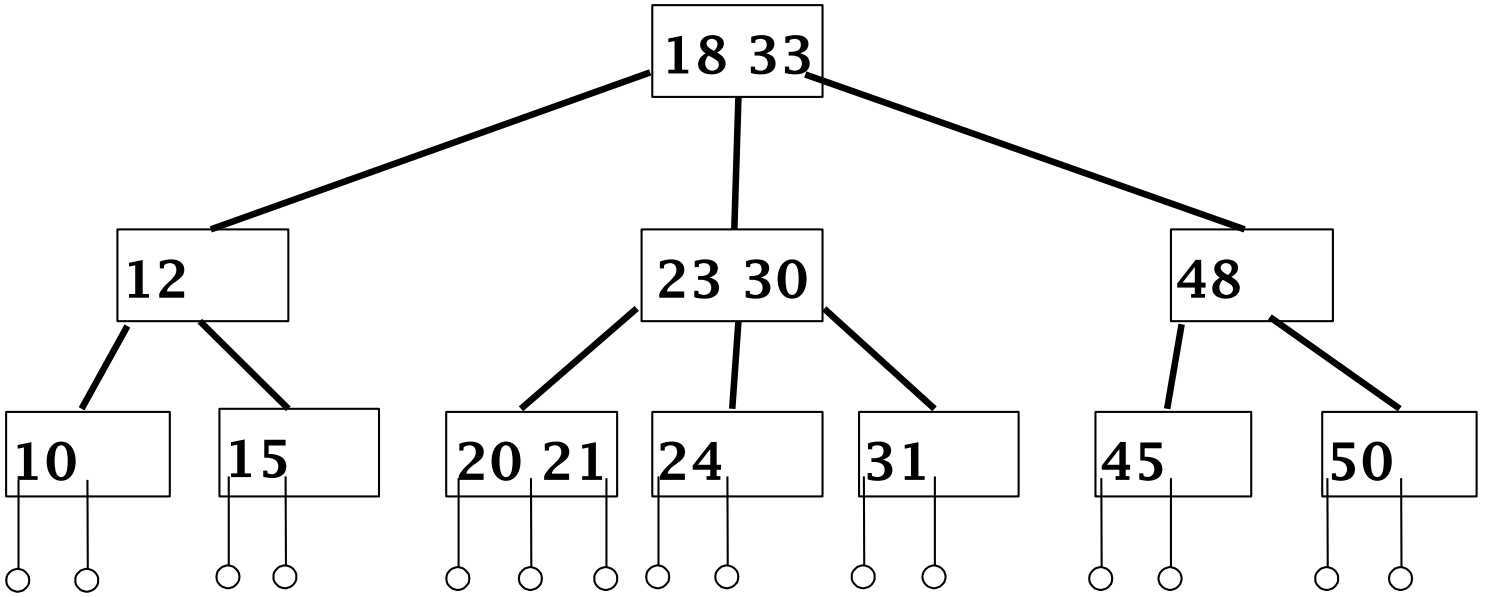
Basic Concepts

- Structure of dynamic indexing
  - The structure would change when inserting/deleting records
- Objective
  - Maintain high efficiency
    - e.g. high searching efficiency

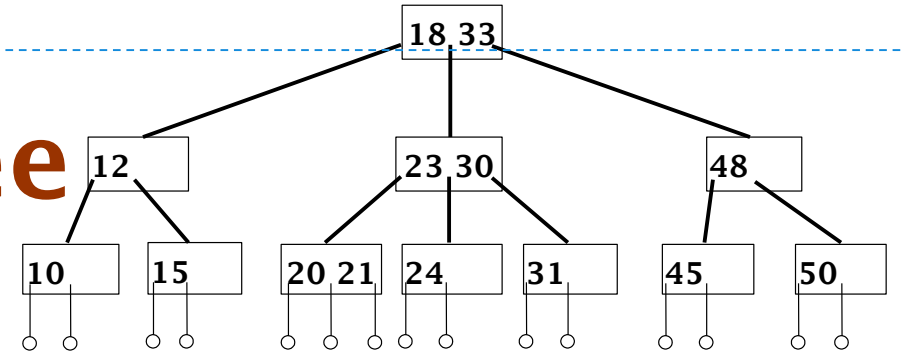
11.4.1 B tree

- ***A type of Balanced Tree***

B tree of order 3 ⇔ 2-3 tree



Structure of m-rank B tree



(1) Each node has at most m children.

(2) Each node other than the root and leaves has at least $\lceil \frac{m}{2} \rceil$ children.

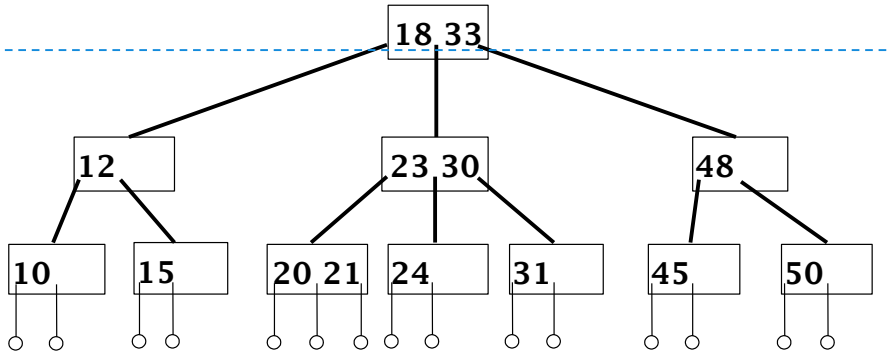
(3) The root has at least two children.

- The only exception is that when the root is the leave node and has no children.
- Then the B tree has only one node.

(4) All leaves are on the same level.

(5) A non-root node having k children has exactly k-1

Properties of B tree



(1) The tree is balanced, and all leaves are on the same level.

(2) Each key value is unique, and the key values in a node are divisions of its children.

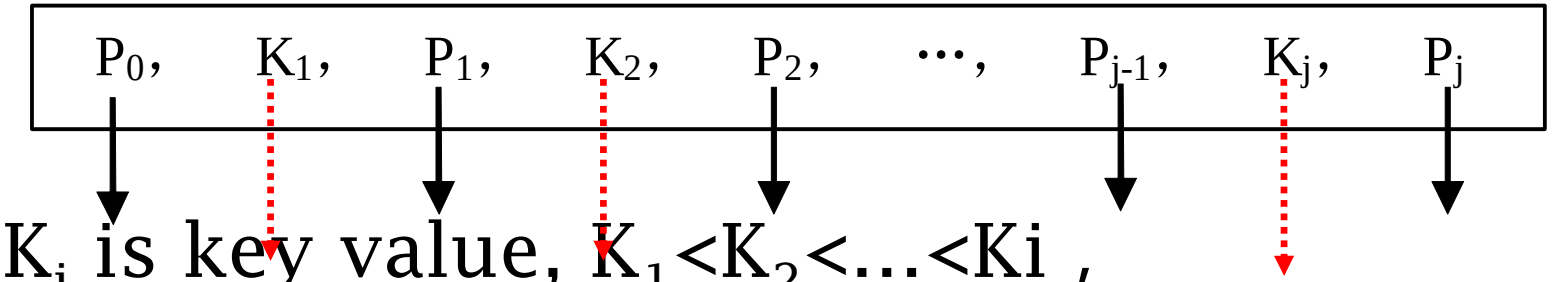
(3) B tree puts (key value) related records on the same page, making use of locality.

(4) It is guaranteed that certain proportion of the B tree is occupied.

- Thus improve the memory utilization.
- Reduce disk visiting times when indexing and updating.

Structure of nodes in B tree

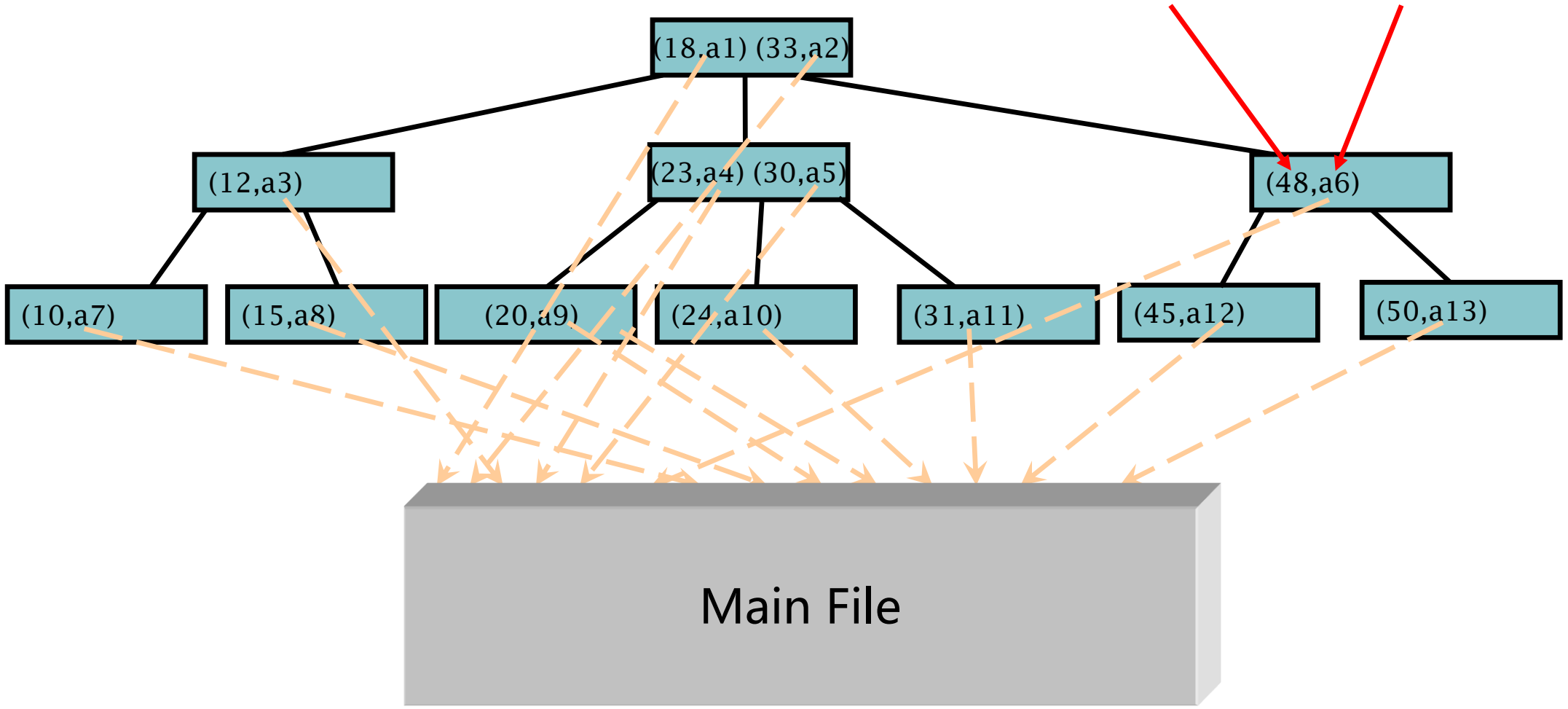
A node with j key values and j+1 pointers in a B tree is usually structured as below :



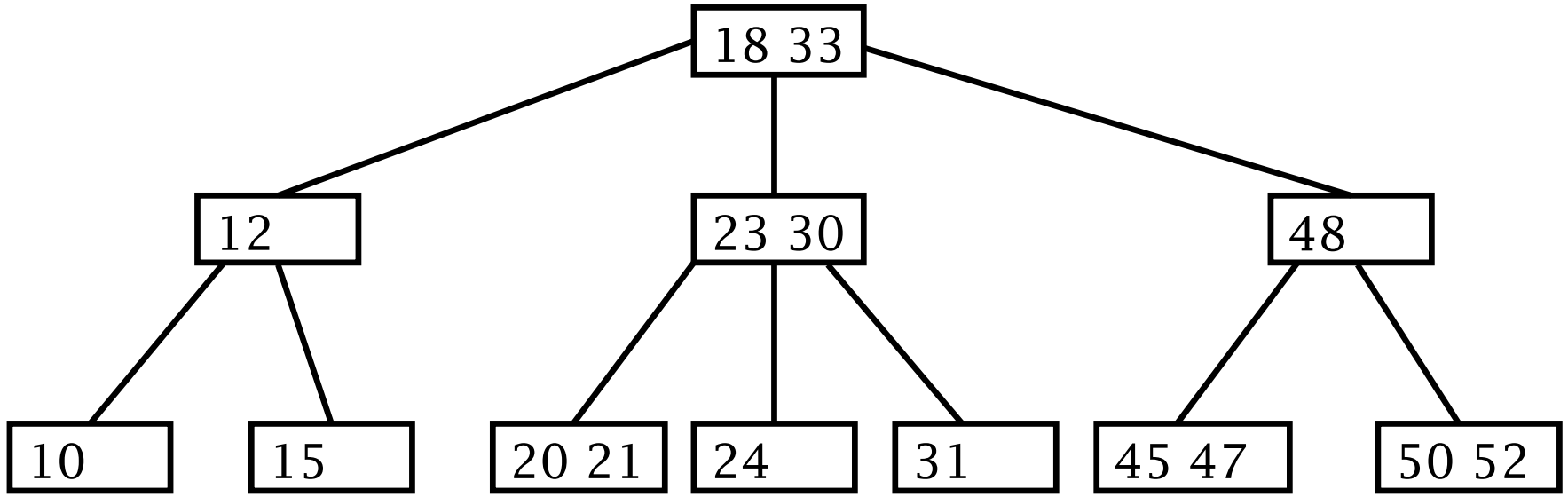
- K_i is key value, $K_1 < K_2 < \ldots < Ki$,
- P_i is a pointer pointing to children nodes with key values between K_i and K_{i+1} .
- Other pointers?

11.4.1 B Tree

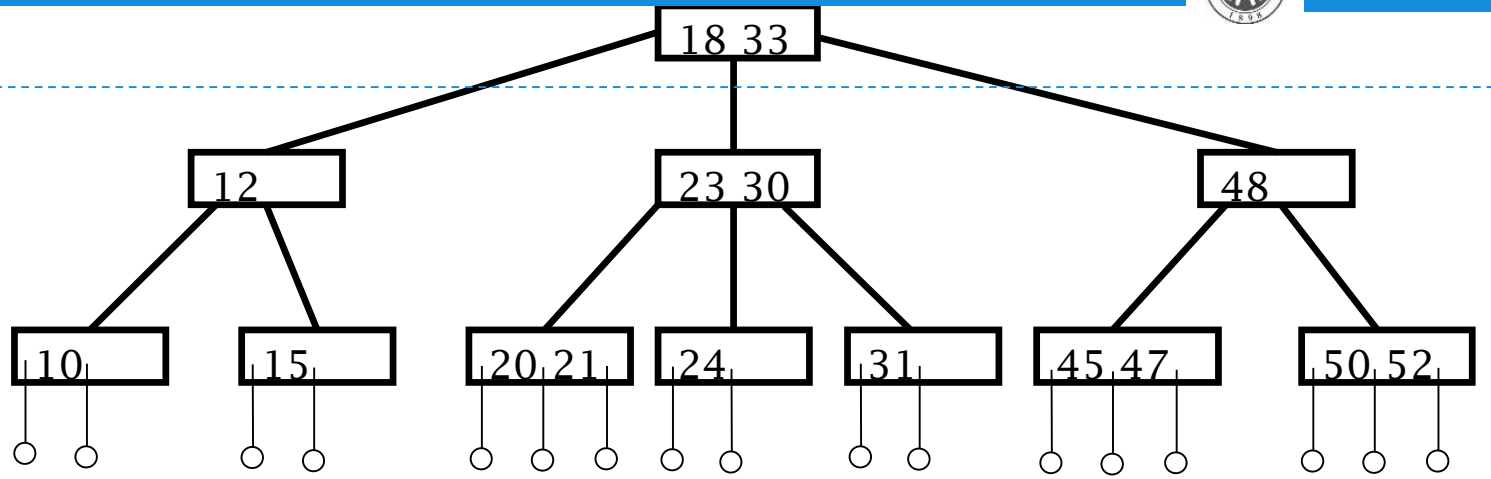
Pointers are implied in B tree (key value, address of the inside file page)



2-3 tree = B tree of order 3



Search in a B tree



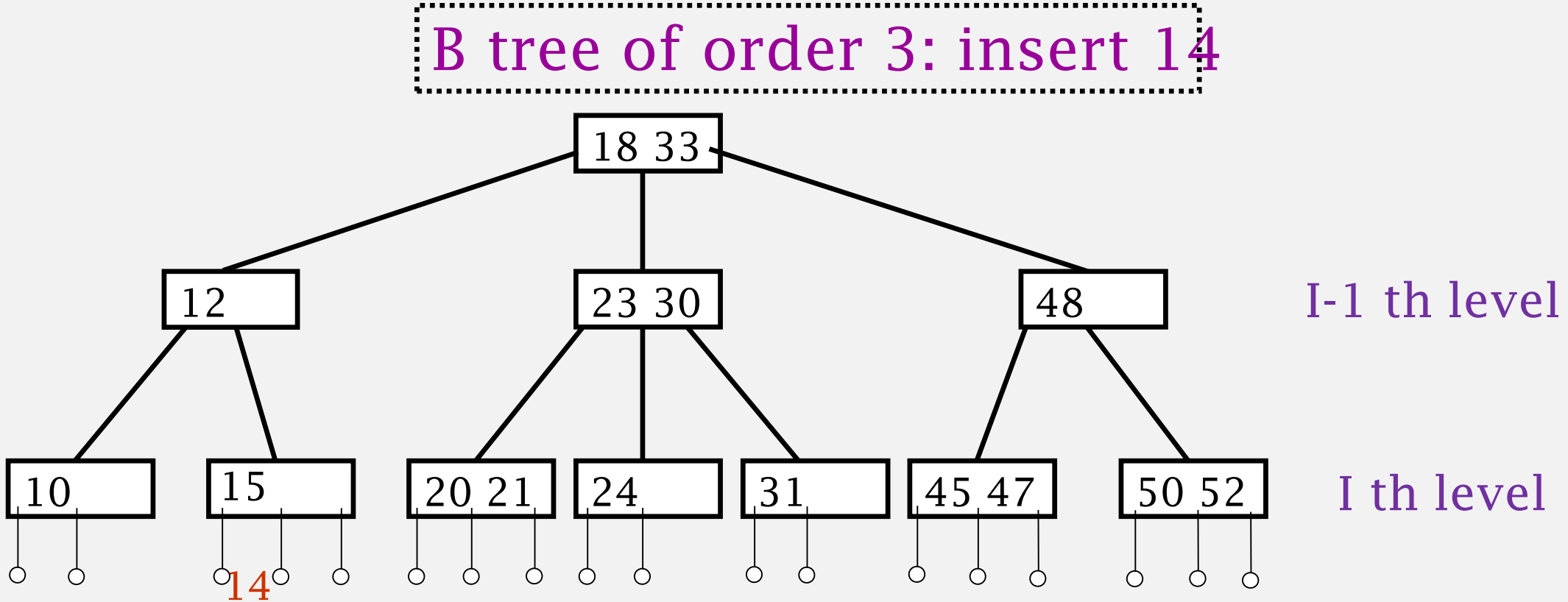
- 2 steps are performed alternately
  - 1. Read the key values of the root, such as K_1, …, Ki, then search the key value in them
    - if it isfound, then the search is completed
  - 2. otherwise, if the key value is between K_i and K_{i+1}, search in the node that p_i points to.
- If p_i is pointing to an external void node, then the searching process is failed.

Insertion in a B tree

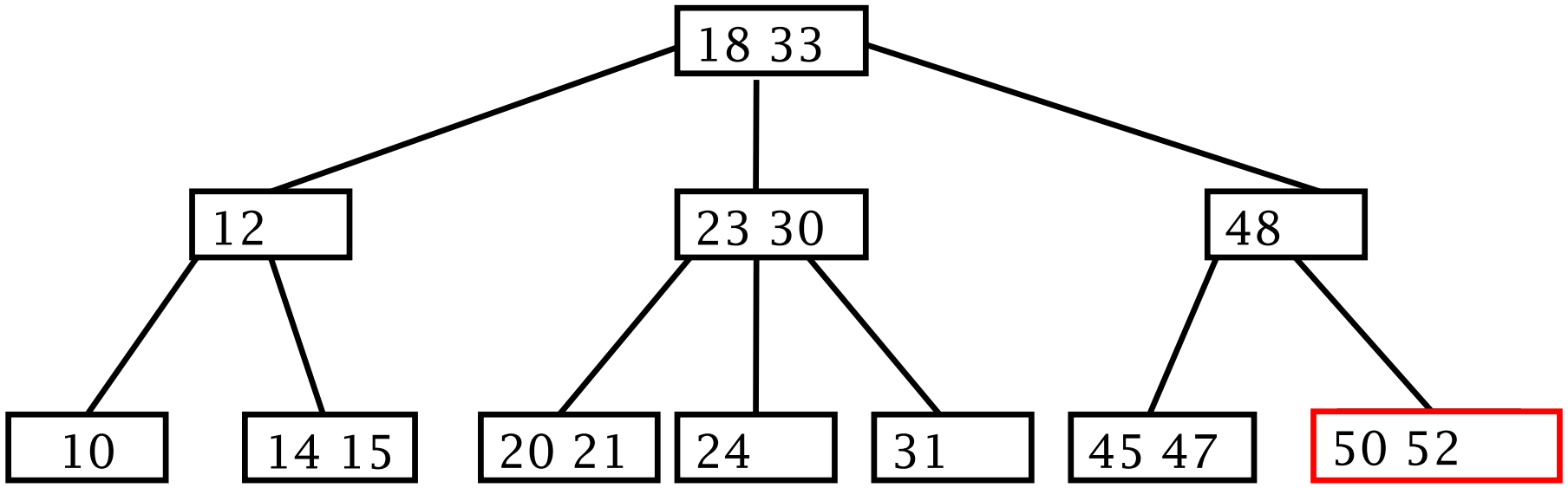
- Need to maintain the properties of B tree, (rank, balance)
  - 1) insert at the lowest level
  - 2) if the current node overflows, then the node needs to be split, and the intermediate key and new pointers are inserted into the father node
  - 3) if the father node also overflows, then the father node also needs to be split
    - Splitting might be transfered to the root (thus increasing the height of the tree)

Insertion in a B tree

- External void node (failed searching) is on the I th level, and the inserted key value is on the I-1 th level

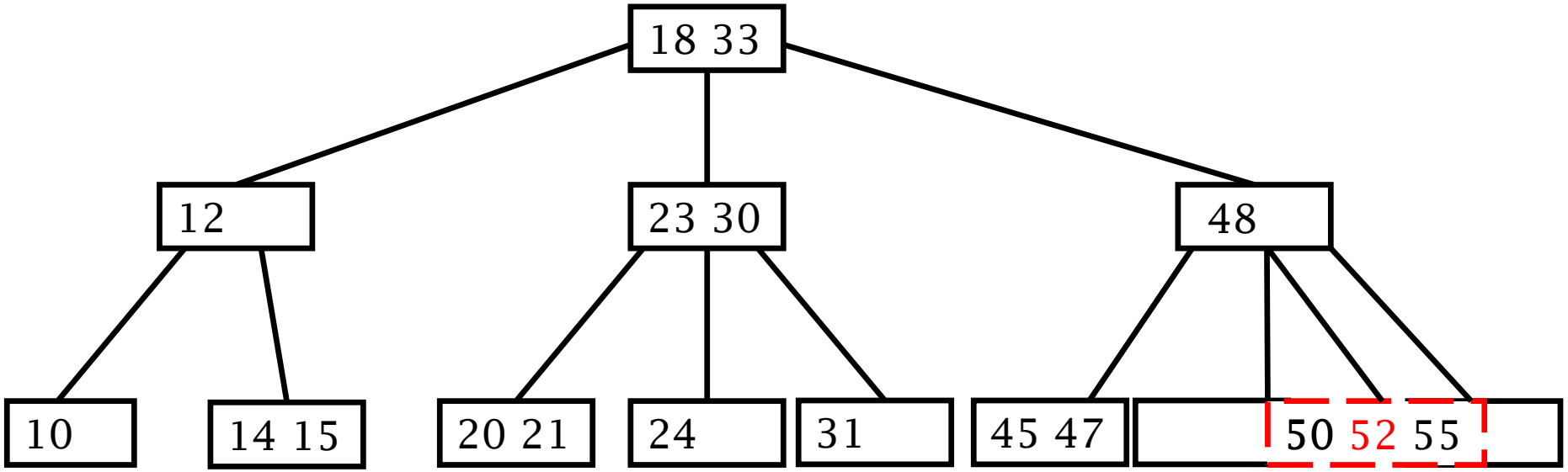


m=3, insert 55

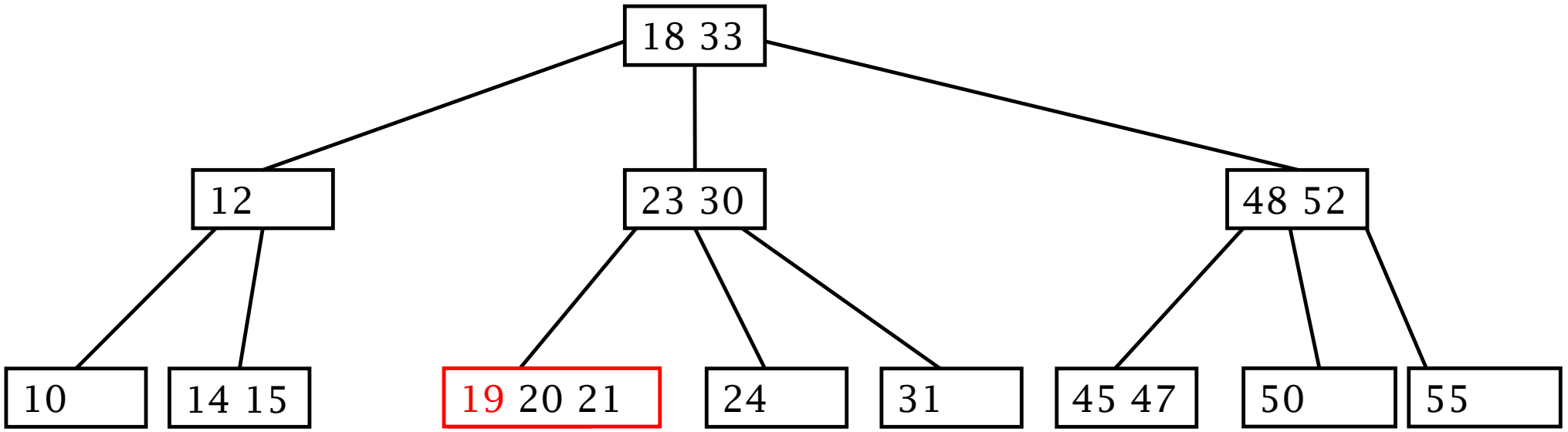


55

m=3, the leaf node split, raise 52 to its father node

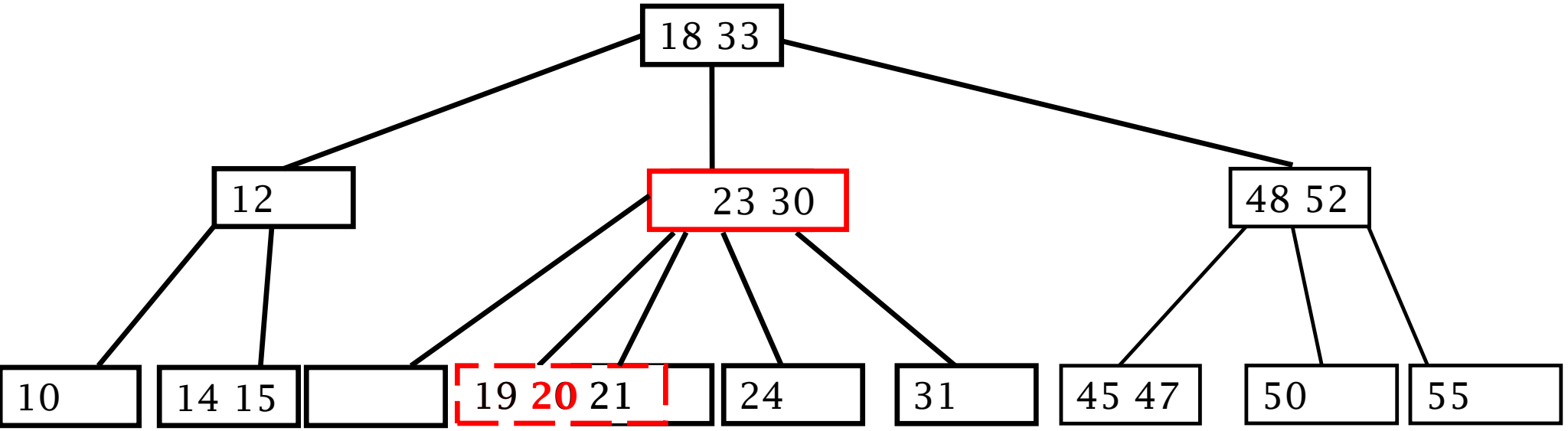


insert 19, causing the root of the B tree of order 3 to split

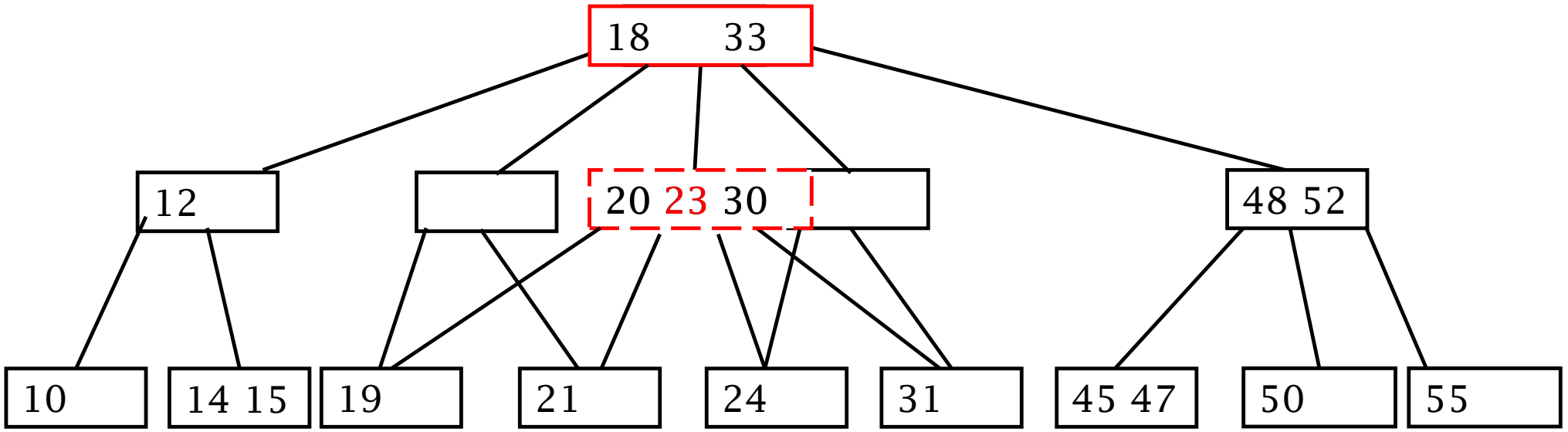


19

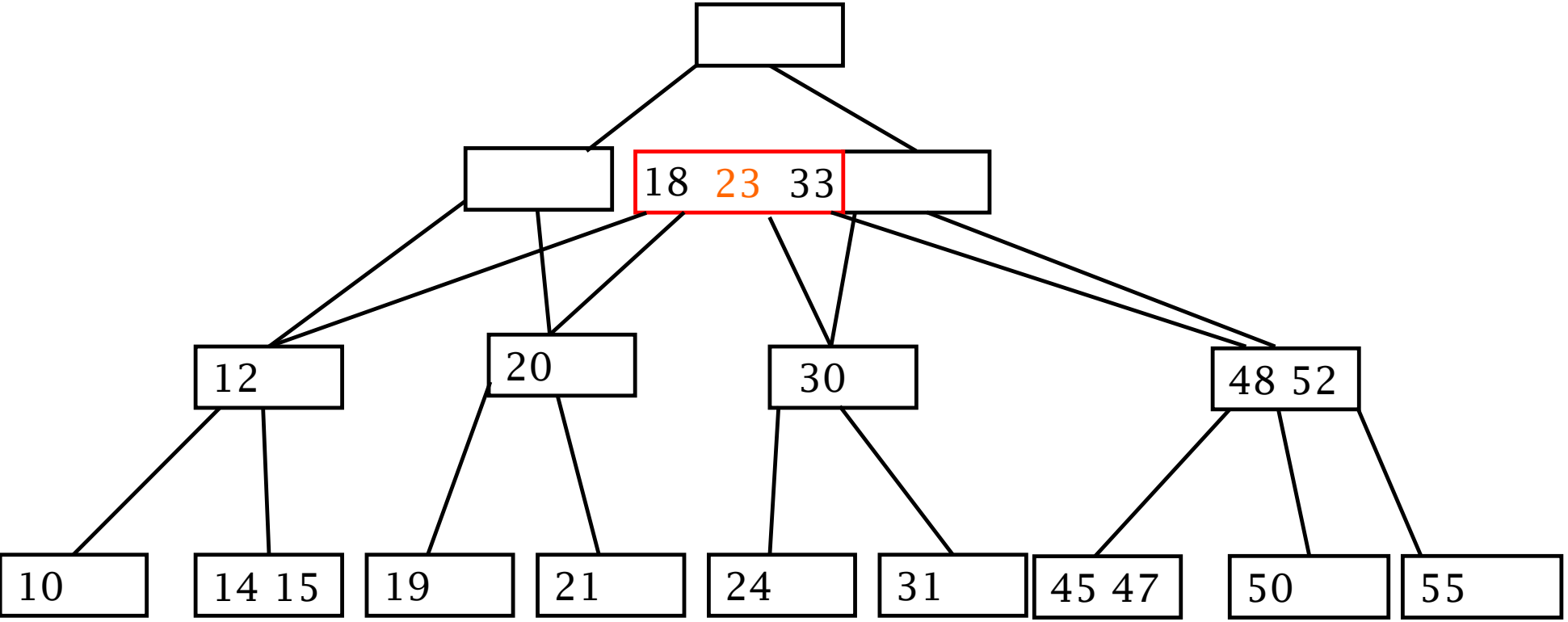
m=3, the leaf splits



m=3, the node on the second level splits

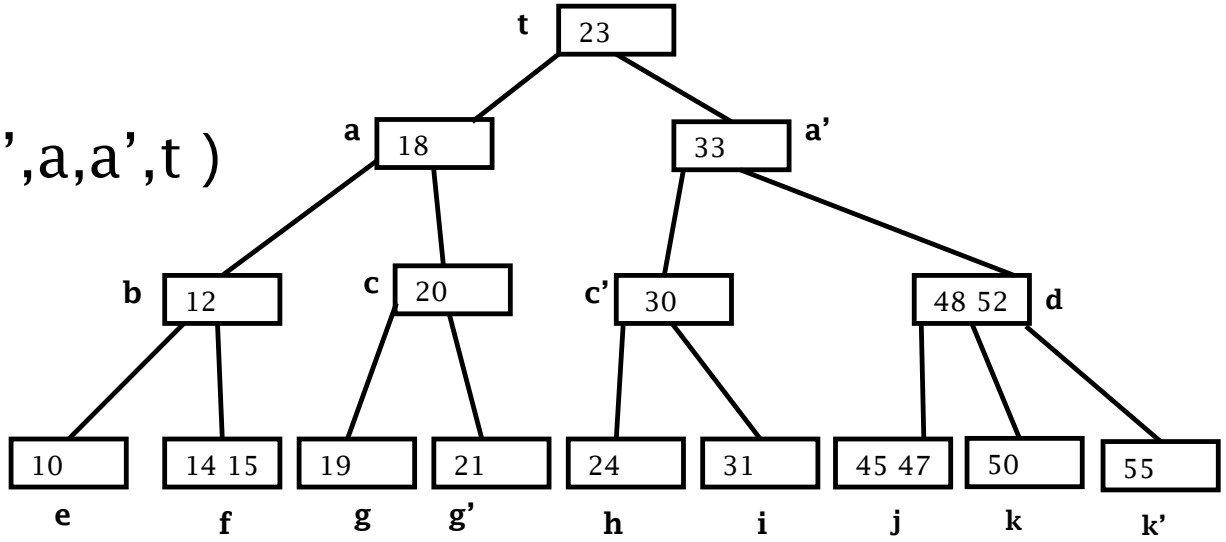


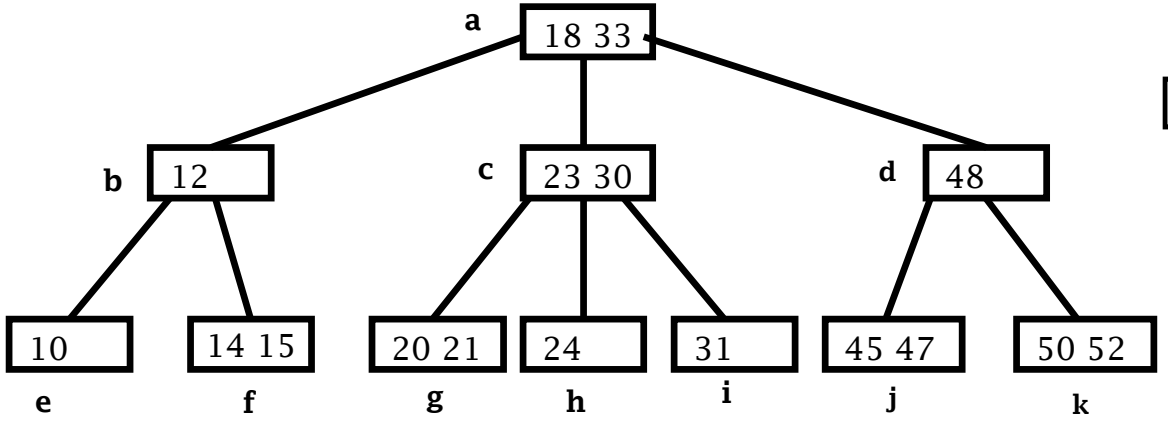
m=3, the root splits



Access time of external memory of B tree

- continuously insert 14, 55, 19, suppose all data accessed have been cached
- read disk 7 times (a,b,f; d,k; c,g)
- write disk 11 times (f; k,k',d; g,g',c,c',a,a',t)

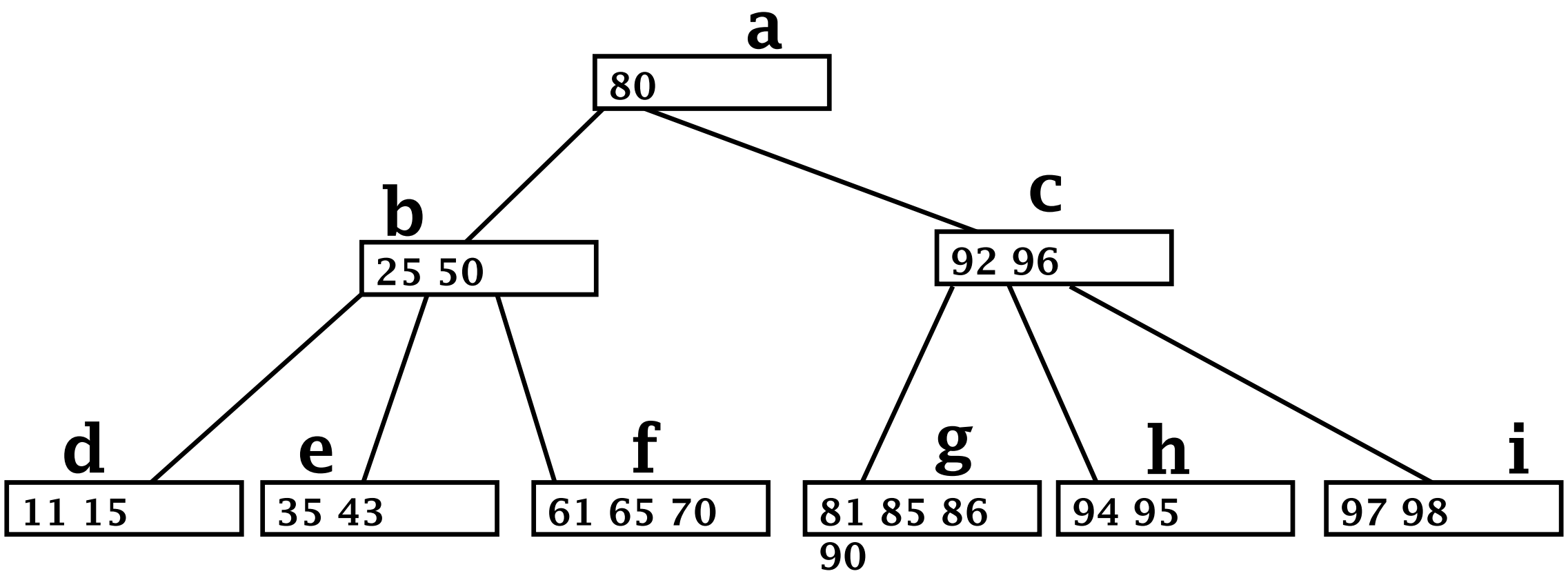




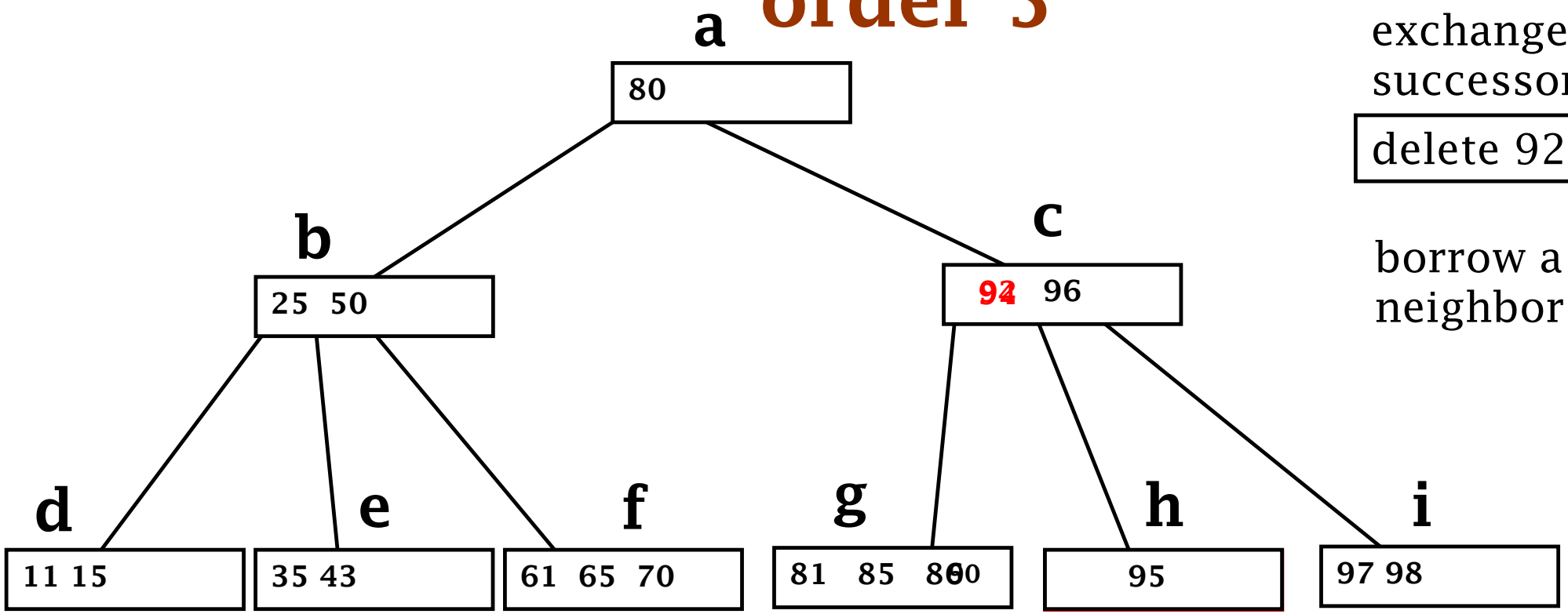
Deletion in a B tree

- If the key value to be deleted is not a leaf, exchange it with its intermediate successor (guaranteed to be a leaf).
- The key value to be deleted is a leaf.
  - if the number of remaining key values after deletion is no less than $\lceil m/2 \rceil - 1$, delete it directly
  - otherwise,
    - if the number of key values of a brother is not equal to $\lceil m/2 \rceil - 1$
      - move some of the brother's key values to the current node. (key values in the father should be changed accordingly)
    - otherwise,
      - merge

e.g. Deletion in a B tree of order 5



e.g., Deletion in a B tree of order 5



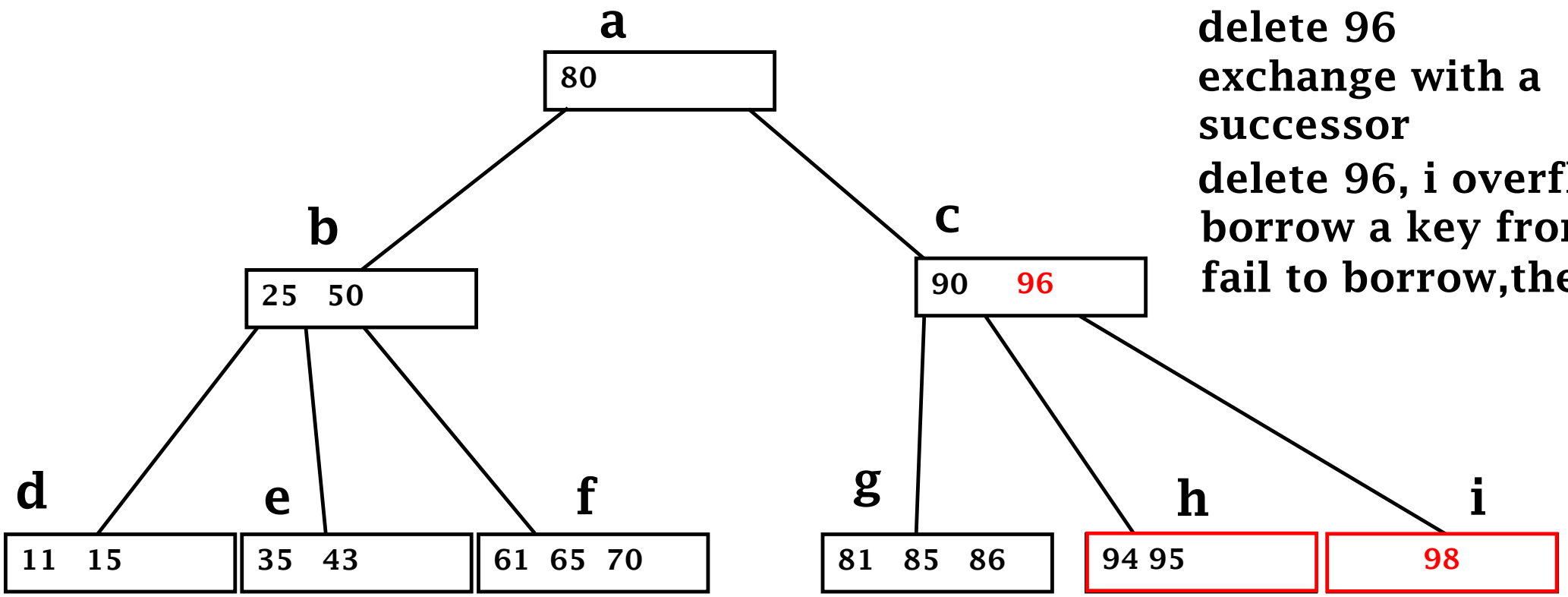
exchange with a successor

delete 92, h overflows

borrow a key from left neighbor

To delete 92, exchange the node with a successor, borrowing a key from its left neighbor

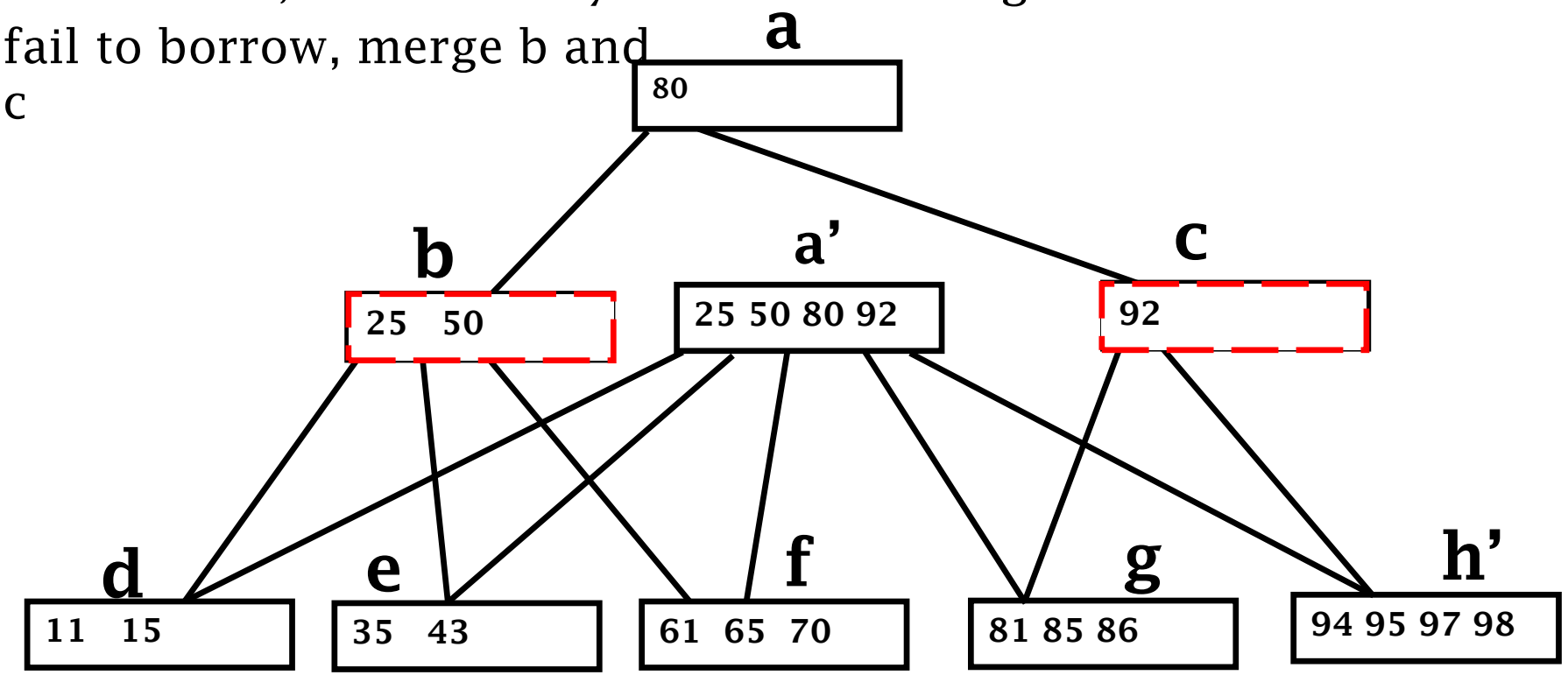
e.g., Deletion in a B tree of order 5



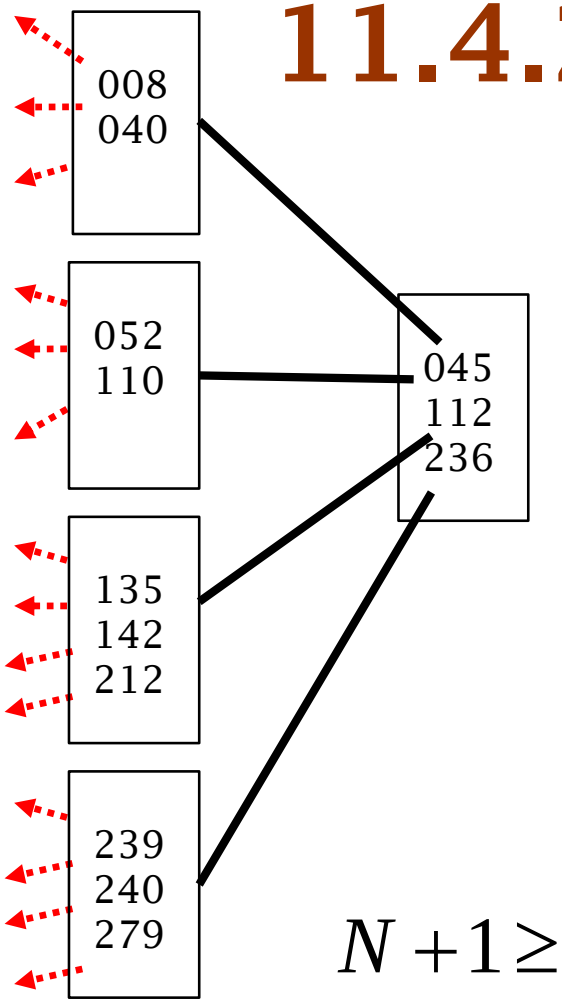
delete 96
exchange with a successor
delete 96, i overflows
borrow a key from left neighbor
fail to borrow,then merge h and i

e.g., Deletion in a B tree of order 5

merge h and i into h'
c overflows, borrow a key from its left neighbor
fail to borrow, merge b and c



11.4.2 Analysis of B tree



- A B tree with N key values
  - has N+1 external null nodes
  - suppose external null nodes are on level k
- number of nodes on each level
  - level 0 is root, level 1 has at least 2 nodes
  - level 2 has at least $2 \cdot \lceil \frac{m}{2} \rceil$ nodes,
  - level k has at least $2 \cdot \lceil \frac{m}{2} \rceil^{k-1}$ nodes,

$$N+1 \geq 2 \cdot \lceil m/2 \rceil^{k-1}, \; k \leq 1 + \log_{\lceil m/2 \rceil}(\frac{N+1}{2})$$

Examples

- N=1,999,998, m=199

$$k \le 1 + \log_{\lceil m/2 \rceil}(\frac{N+1}{2})$$

  - k=4
  - visit 4 levels at most for searching once

Times of splitting

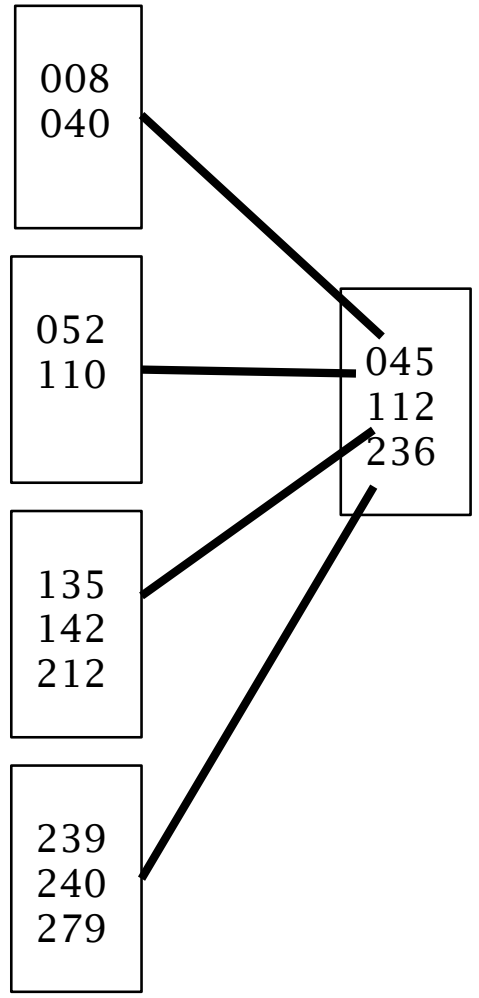
- Suppose there are N keys (N+1 void pointers), p internal nodes

  so, $N \geq 1+(\lceil m/2 \rceil -1)(p-1)$

- In the worst case, each inserting involves splitting (except the first one), i.e., each node is formed by splitting, then the average number of nodes split when inserting a key is:

$$p-1 \leq \frac{N-1}{\lceil m/2 \rceil -1}$$

$$s=\frac{p-1}{N} \leq \frac{N-1}{(\lceil m/2 \rceil -1)\cdot N} \leq \frac{1}{\lceil m/2 \rceil -1}$$



Discussion

- 1. Is there a B tree of order 2 that conforms to definitions of B tree? Is it of any pratical use? Why?
- 2. When deleting a node from an B tree, we consider borrowing before merging, so why not consider giving keys to brothers before splitting?
- 3. In the definition of B tree, degree ranges from $\lceil \frac{m}{2} \rceil$ to m , is it possible to alter this?

Data Structures and Algorithms

Thanks

the National Elaborate Course (Only available for IPs in China)
http://www.jpk.pku.edu.cn/pkujpk/course/sjjg/
Ming Zhang, Tengjiao Wang and Haiyan Zhao
Higher Education Press, 2008.6 (awarded as the "Eleventh Five-Year" national planning textbook)