B Trees

- Commonly used for organizing data on disk
  - Disk access time (time to read or write a block) dominates cost
  - Very important to minimize number of disk accesses
- Some notes on terminology:
  - This textbook uses the name *B tree*, but elsewhere they are known as *B+ trees*
  - In this textbook, the *order* of a B tree is the maximum number of children per node.
  - Elsewhere, *order* refers to the minimum number of children in index nodes other than the root.

B tree definitions

A B-tree of order M is an M-ary tree such that:

1. Data items are only in the leaves
2. Non-leaf (index) nodes store up to M-1 *keys*:
   - key *i* determines the smallest possible key in subtree *i*+1.
3. The root is either a leaf or has between 2 and M children

Every node other than the root is at least half-full:

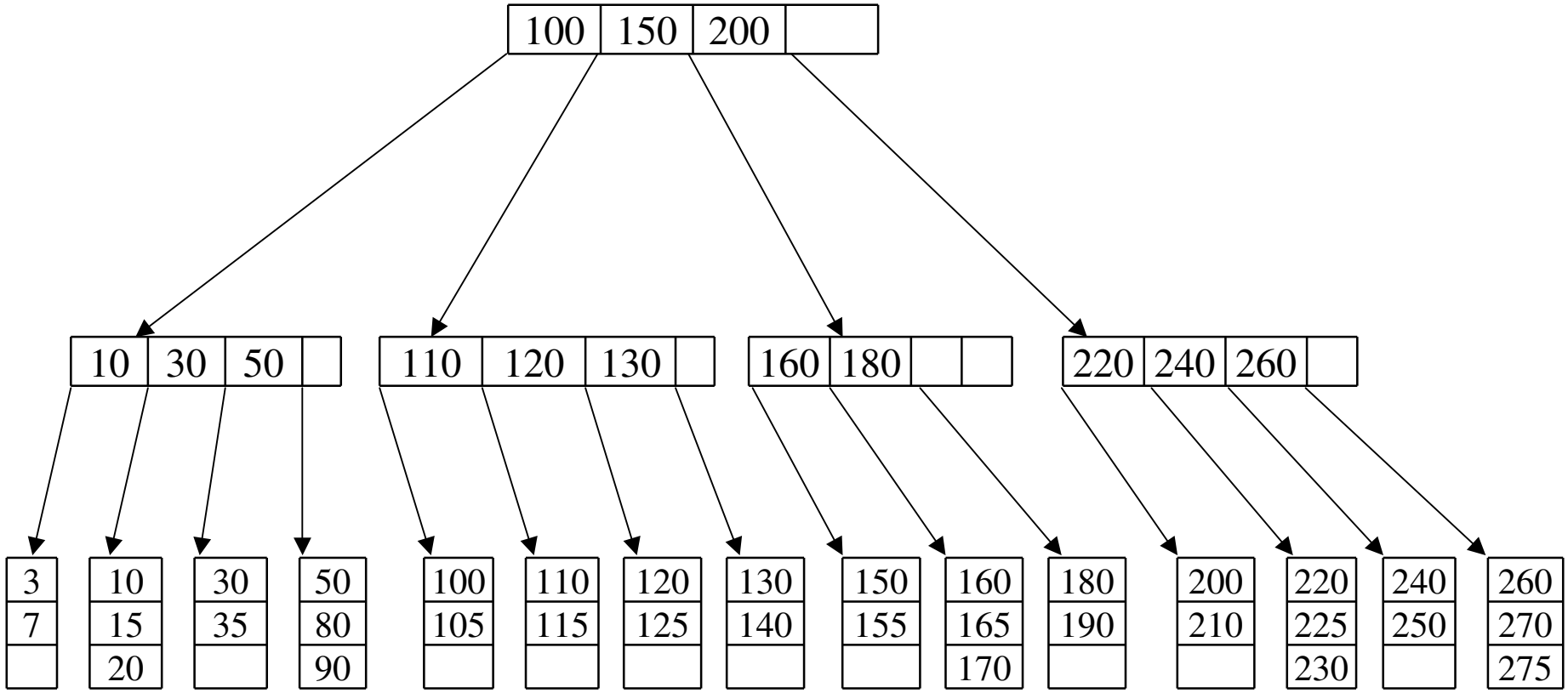
4. All non-leaf nodes (except root) have at least $\lceil M/2 \rceil$ children
5. All leaves are at the same depth and have between $\lceil L/2 \rceil$ and L records.

B tree and Binary Search Tree comparison

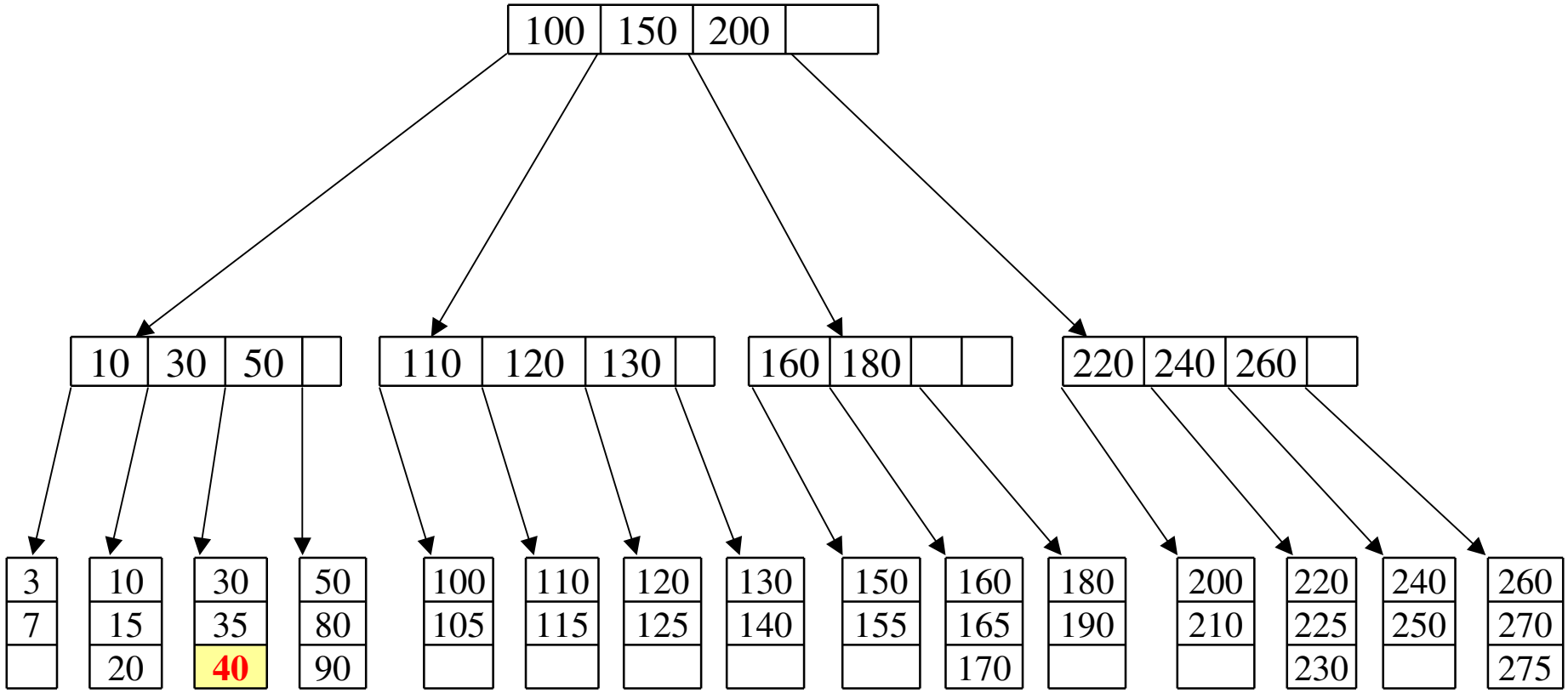
- **Binary search trees:** nodes have 0, 1 or 2 children and 1 key
- **B-trees:** non-leaf nodes have up to M children: a node with d keys has d+1 children
- **Binary search trees:** data is stored in both leaf and non-leaf nodes, and for any given index node N:
  - N's left subtree contains only items with keys < N's key
  - N's right subtree contains only items with keys > N's key
- **B-trees:** non-leaf nodes contain up to M-1 keys (k_1, …, k_{M-1}):
  - Subtree to left of k_1 contains only items with keys < k_1
  - Subtree between k_i and k_{i+1} contains only items with keys $<k_{i+1}$ and $\geq k_i$
  - Subtree to right of k_{M-1} contains only items with keys $\geq k_{M-1}$

B-tree Example

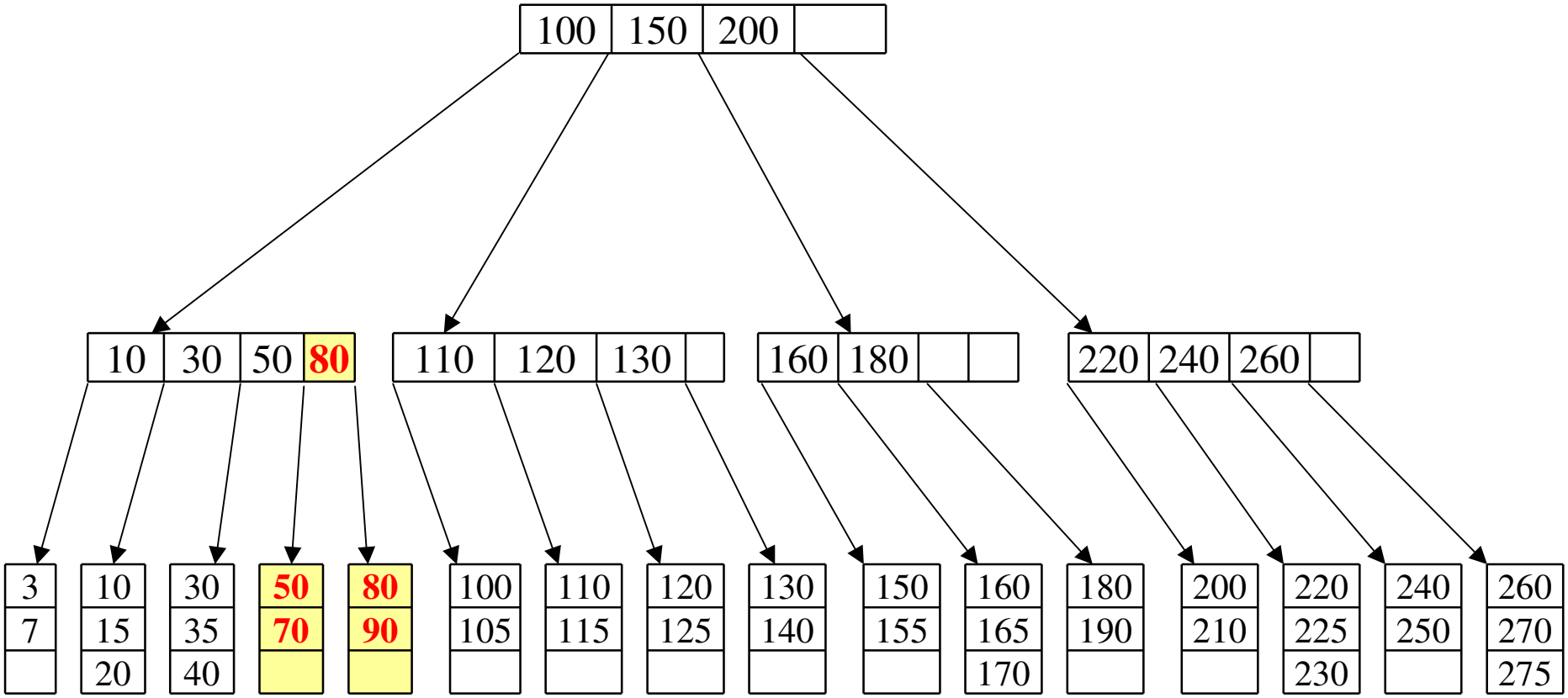
M=5 and L=3



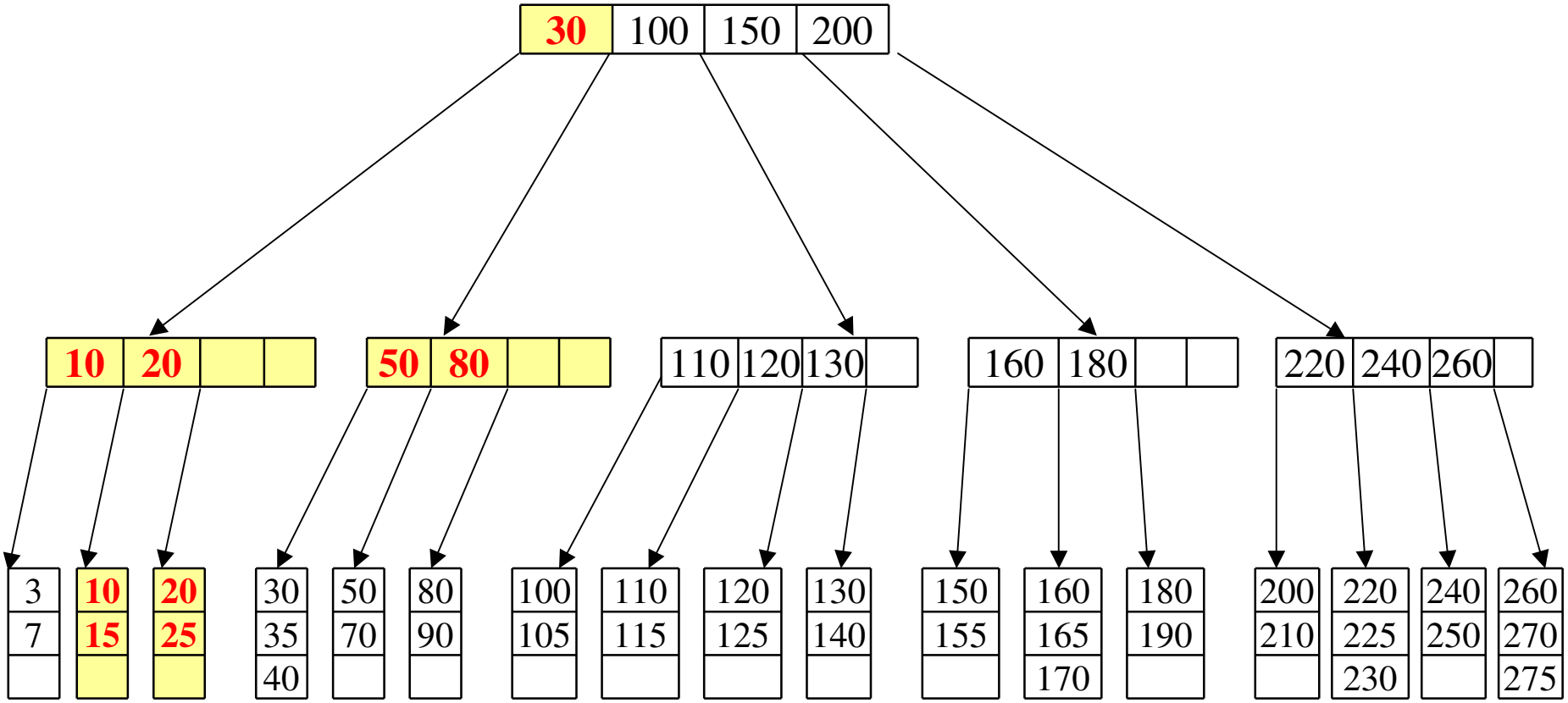
B-tree example: insert 40



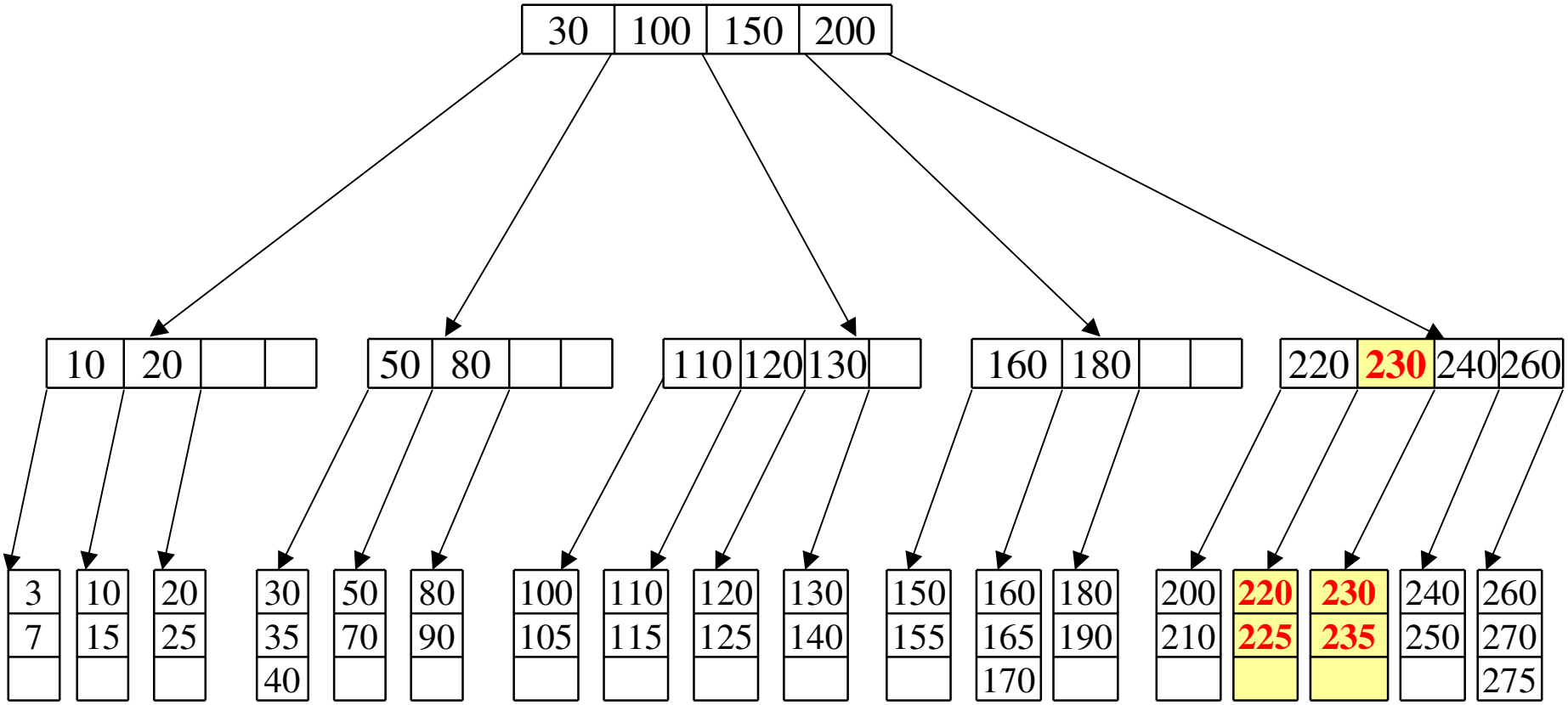
B-tree example: insert 70



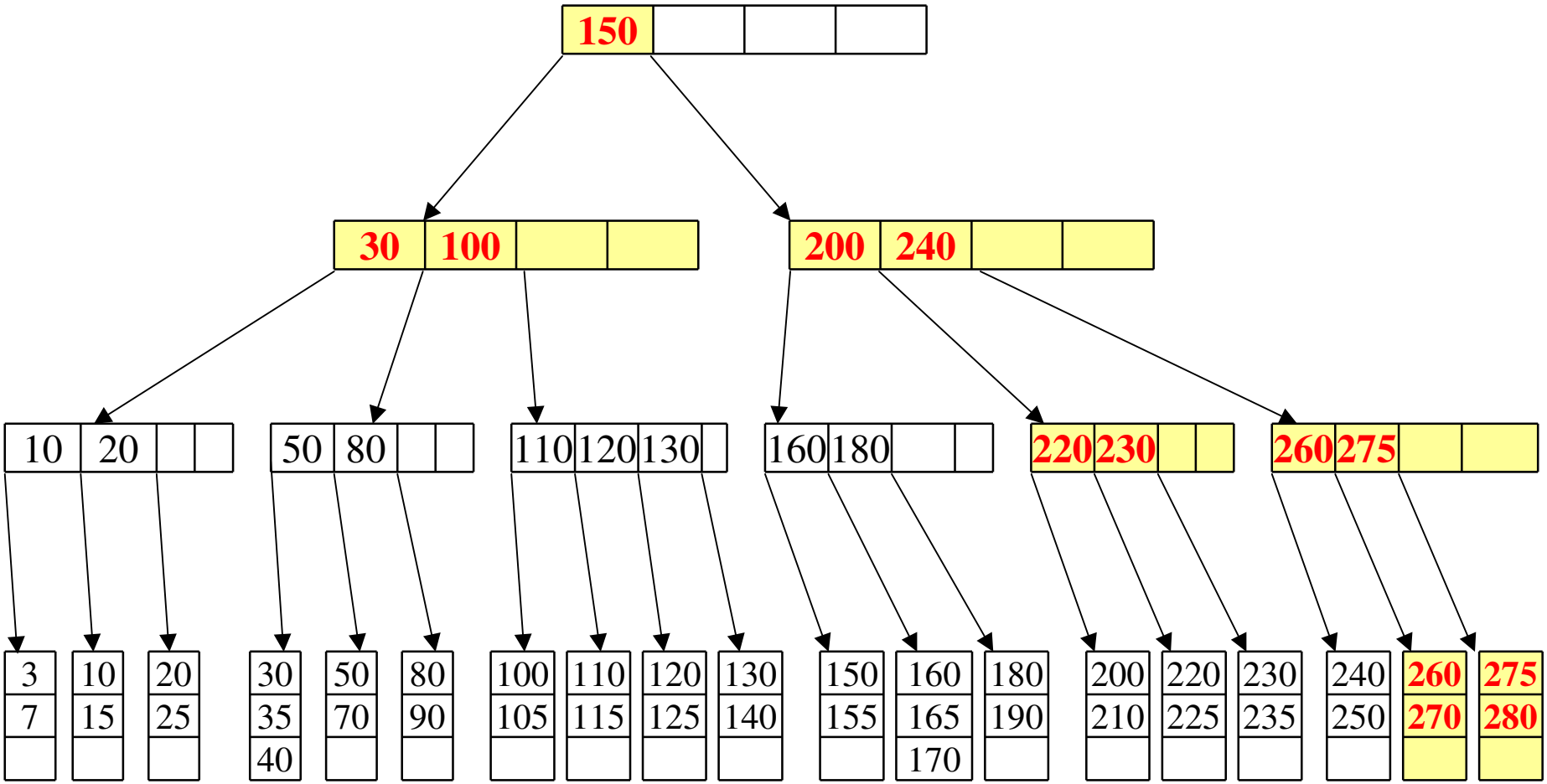
B-tree example: insert 25



B-tree example: insert 235



B-tree example: insert 280



Heaps

A heap is a binary tree that satisfies all of the following properties:

- **Structure property:** It is a complete binary tree
- **Heap-order property:** Each node satisfies the *heap condition:*
  - The key of any node must be smaller than (or equal to) the keys of its children, i.e.
    `n.info` ≤ `n.left.info` and
    `n.info` ≤ `n.right.info`, for all nodes n