

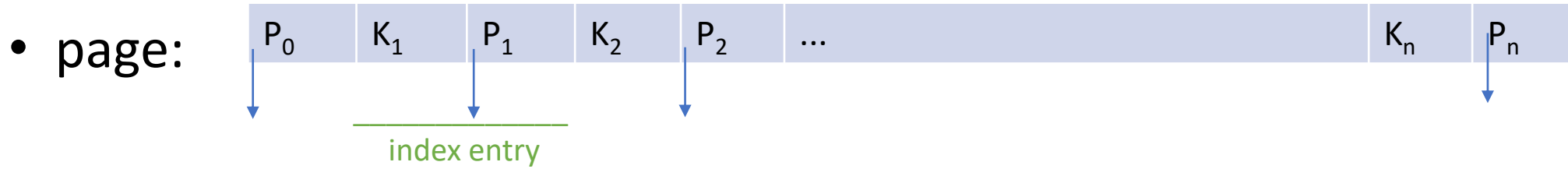
Databases

Indexed Sequential Access Method (ISAM)*

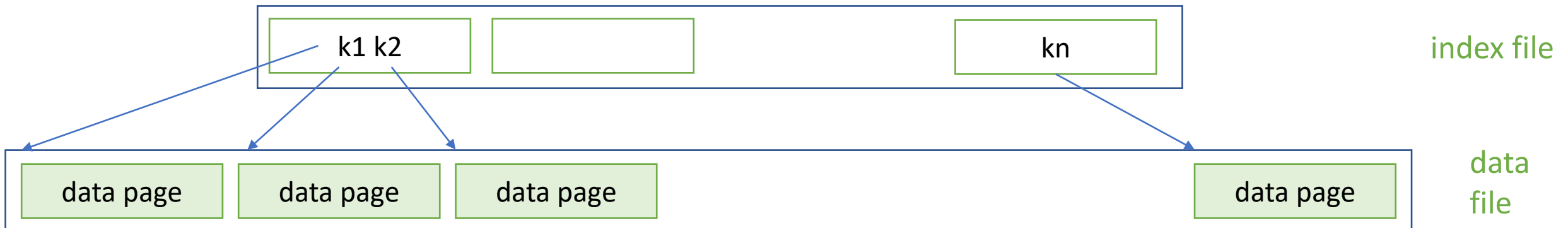
* extra material: not required for the exam

* Example. Q: Find all phones with *rating* > 9 - range selection query

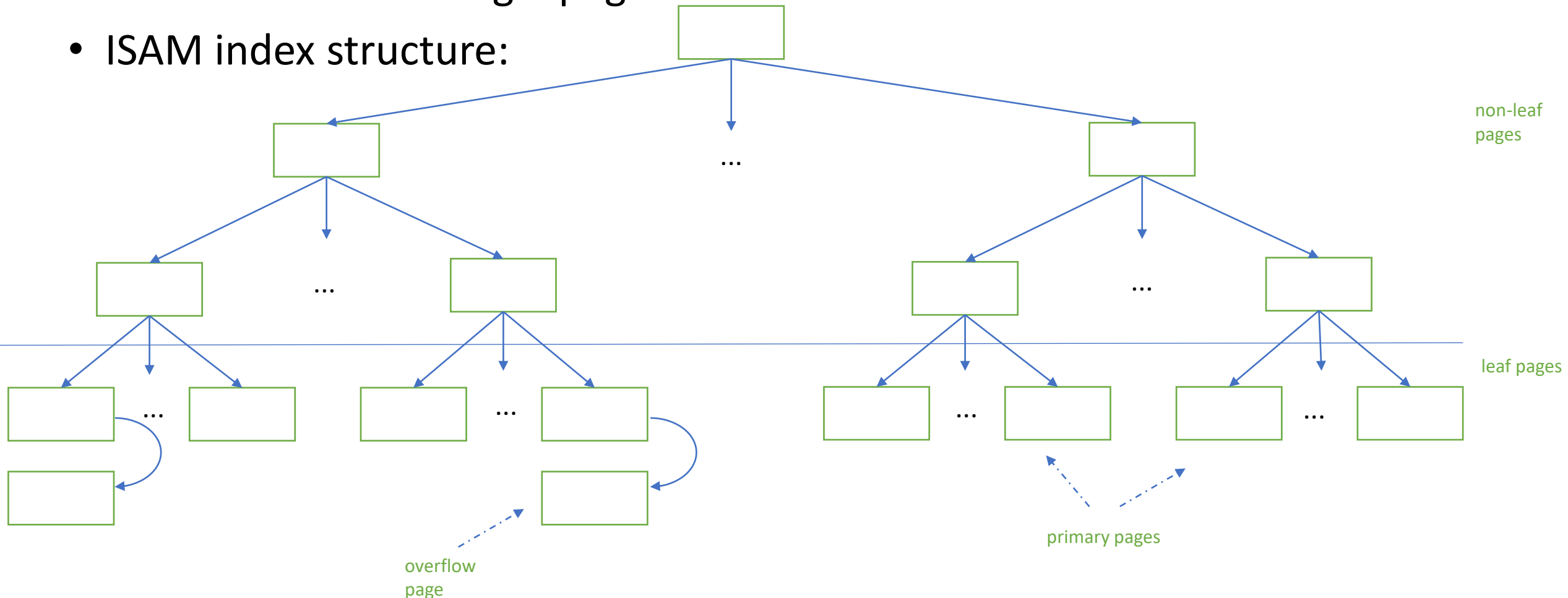
- data stored in sorted file (records sorted by *rating*) - identify 1st phone using binary search; scan file to get the rest of the phones
- large file => potentially expensive binary search
- create another file with records of the form $\langle 1^{st} \text{ key on the page, pointer to the page} \rangle$, sorted on the key (*rating* in the example)



- one-level index structure:



- size of index file - much smaller than size of data file => faster binary search
- index file can still be quite large => further optimization: auxiliary structures are created recursively on top of previously created ones, until one such structure fits on a single page
- ISAM index structure:



- file creation
 - allocate leaf pages - sequentially allocated, sorted on the key
 - allocate non-leaf pages
 - inserts that exceed a page's capacity – allocate overflow pages
- search
 - starts at the root
 - comparisons with the key to find the leaf page
 - cost – disk I/O

data pages

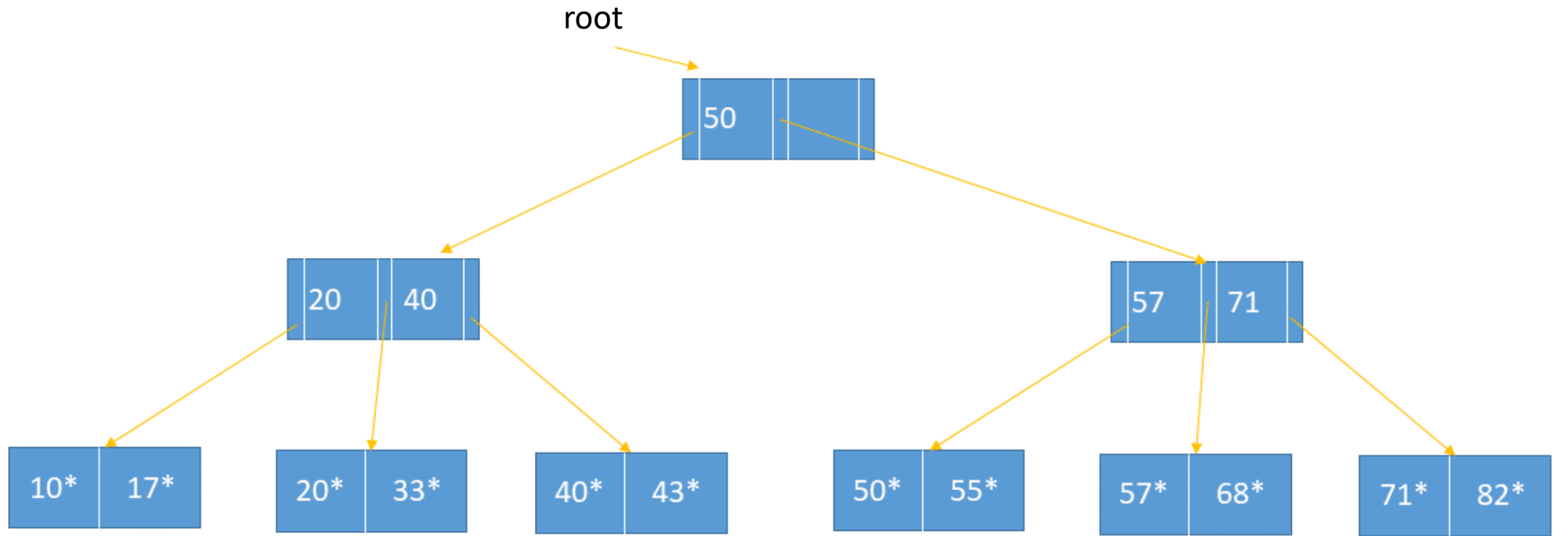
index pages

overflow pages

- insertion
 - find the corresponding leaf page, add the entry
 - if there is no space on the page, add an overflow page
- deletion
 - find the leaf page that contains the entry, remove the entry
 - if an overflow page is emptied, it can be eliminated
- inserts / deletes
 - only leaf pages are affected (static structure)

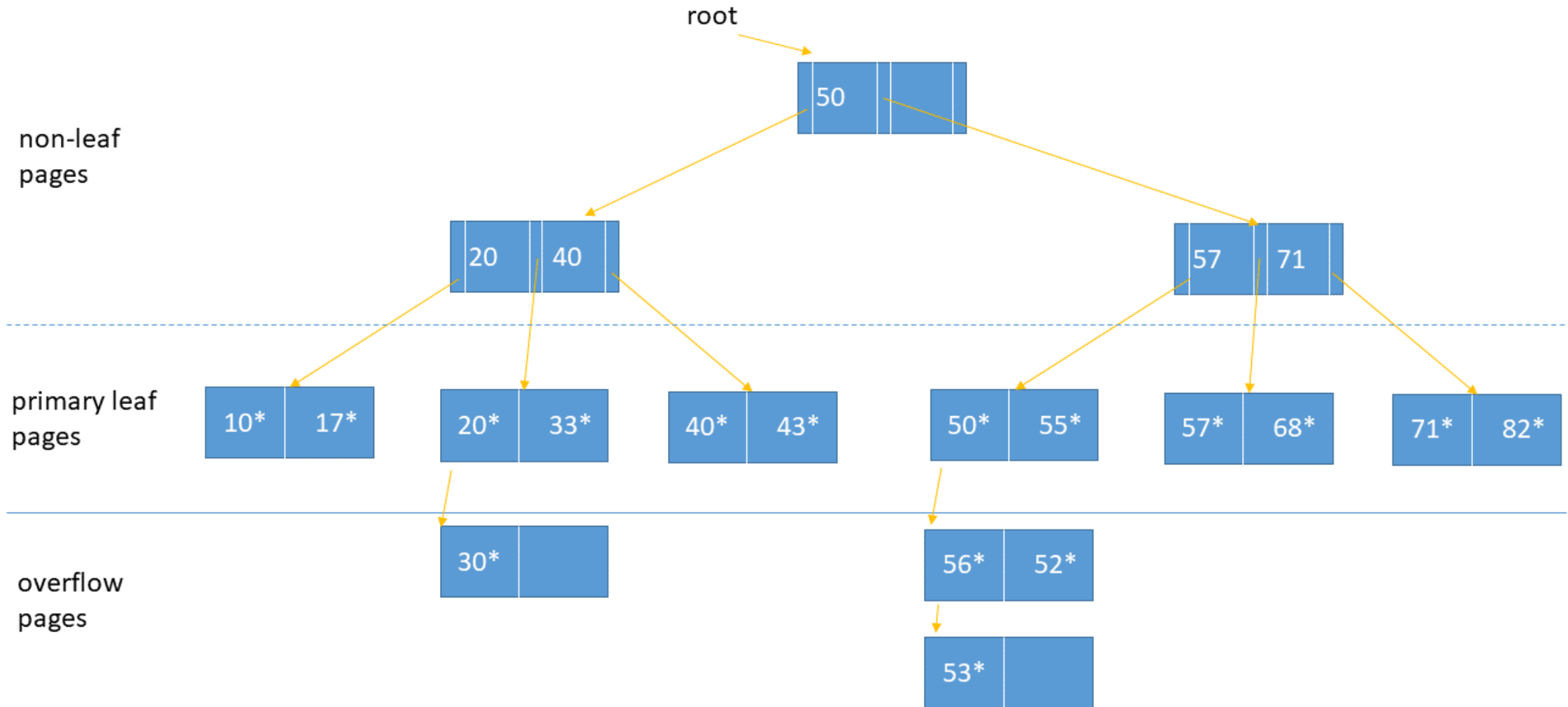
* Example - ISAM tree

- leaf page - 2 entries

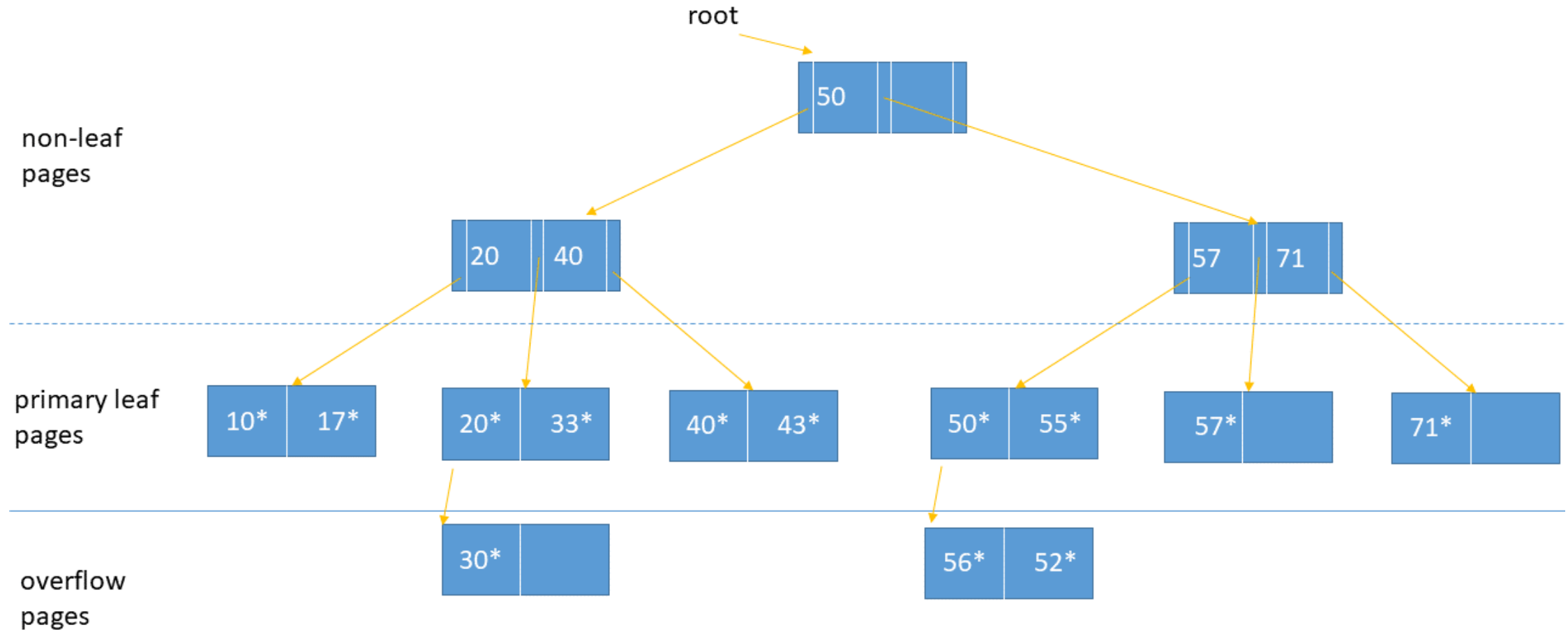


- only key values are shown

- after inserting 30*, 56*, 52*, 53*



- after deleting 53*, 68*, 82*



- benefits and drawbacks
 - better concurrent access, since only leaf pages are modified
 - long overflow chains can develop
 - usually not sorted (to optimize inserts)
 - irregular search time if structure not balanced
 - eliminated through deletes / file reorganization
 - when creating the tree - 20% of each page free for future inserts
 - ISAM - suitable when data size / distribution are relatively static