Washington University in St. Louis

Engineering

Database Management Systems

- B+ Trees

Doug Shook

Structure

- Internal Nodes
- Leaf Nodes
- How is this an advantage over b-trees?

Key Terms

- K is the value being searched for / inserted / deleted
- b is called the fanout
  - What dictates this?

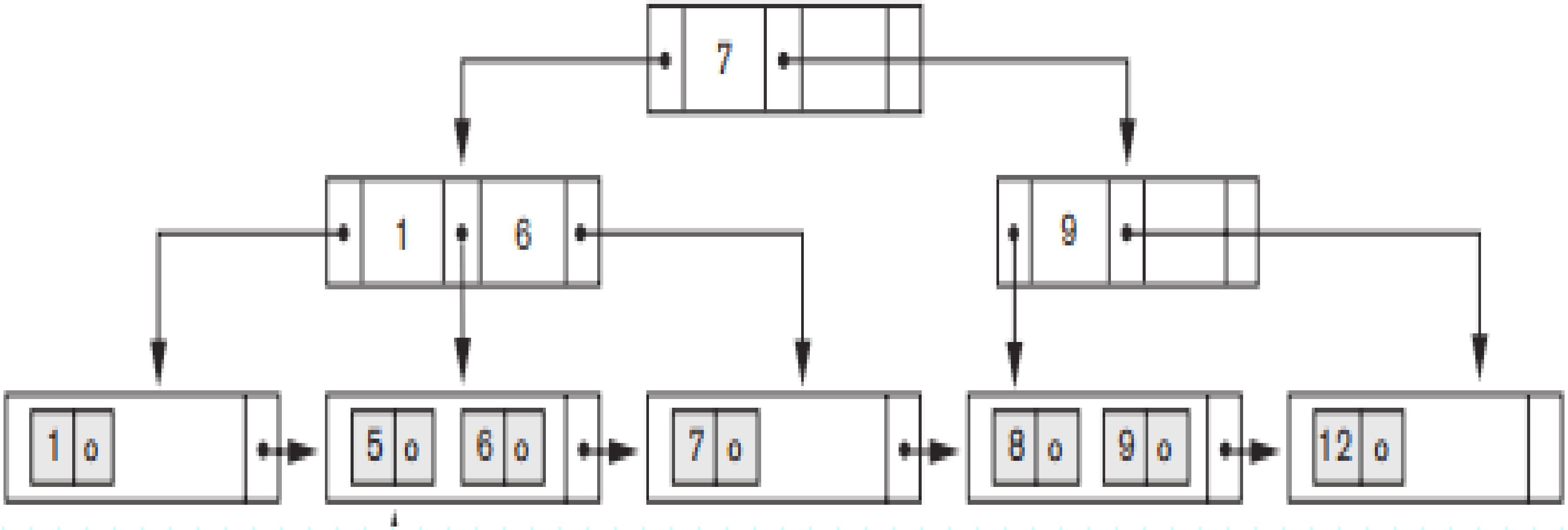
B+-Tree Properties

- The number of elements in a node is dictated by the degree of the tree (p):
  - Internal Nodes (children): ciel(p/2) → p
    - Exception: root
    - Minimum number of search keys?
  - Leaf nodes (keys): ciel(p/2) → p
- The degree of the inner nodes and leaf nodes does not have to be the same

Searching

```
Function search(k, node):
    if node is a leaf
        return node
    else
        if $k < k_0$
            return search(k, $p_0$)
        else if $k > k_q$
            return search(k, $p_q$)
        else
            Find value $k_i$ such that $k_i < k < k_{i+1}$
            return search(k, $p_{i+1}$)
```

Search Example

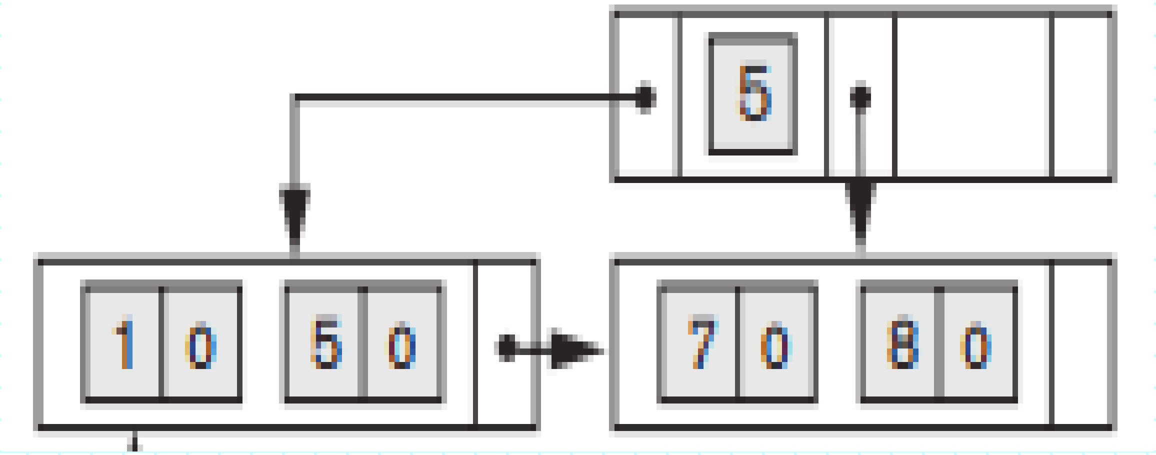


Insertion

```
Search for the node where the new record should go
If the target node is not full, add the record
else:
    Make a new node that contains half the values of the old one
    Insert the largest key of the new node into the parent
    If the parent is full:
        Split the parent and add the middle key to its parent
        Repeat until a split is not needed
    If the root needs to split:
        Create a new root with one key and two pointers
```

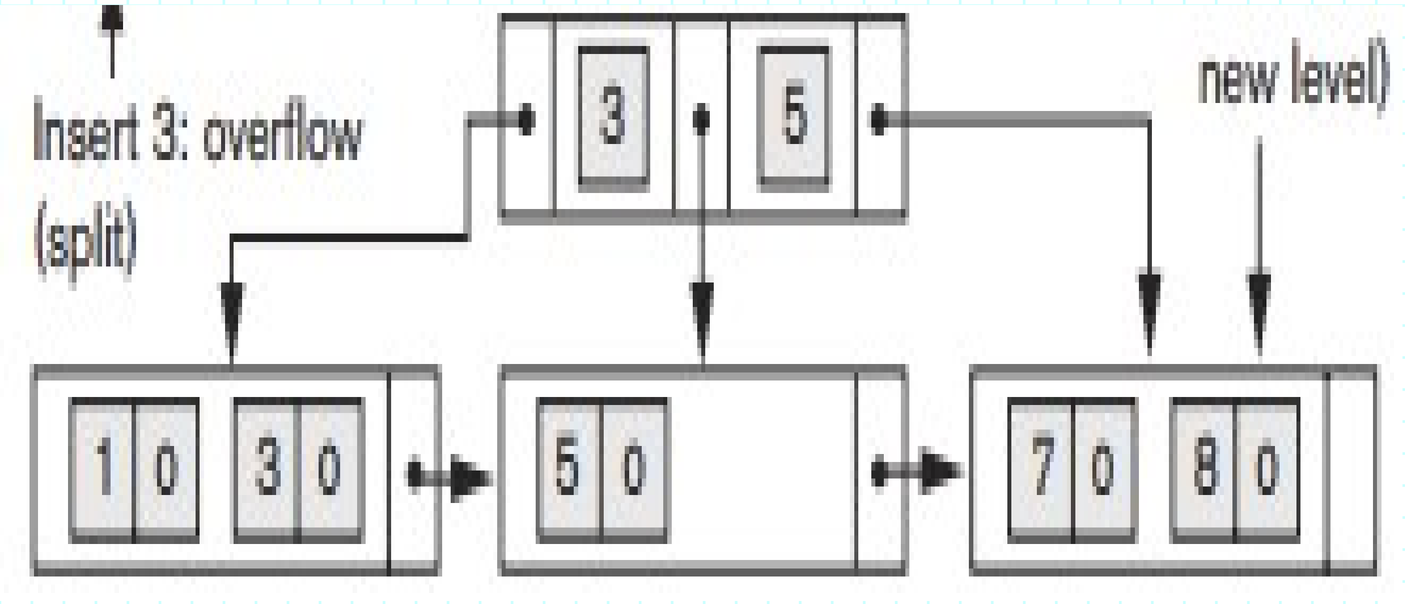
Insertion

Insert a 3:



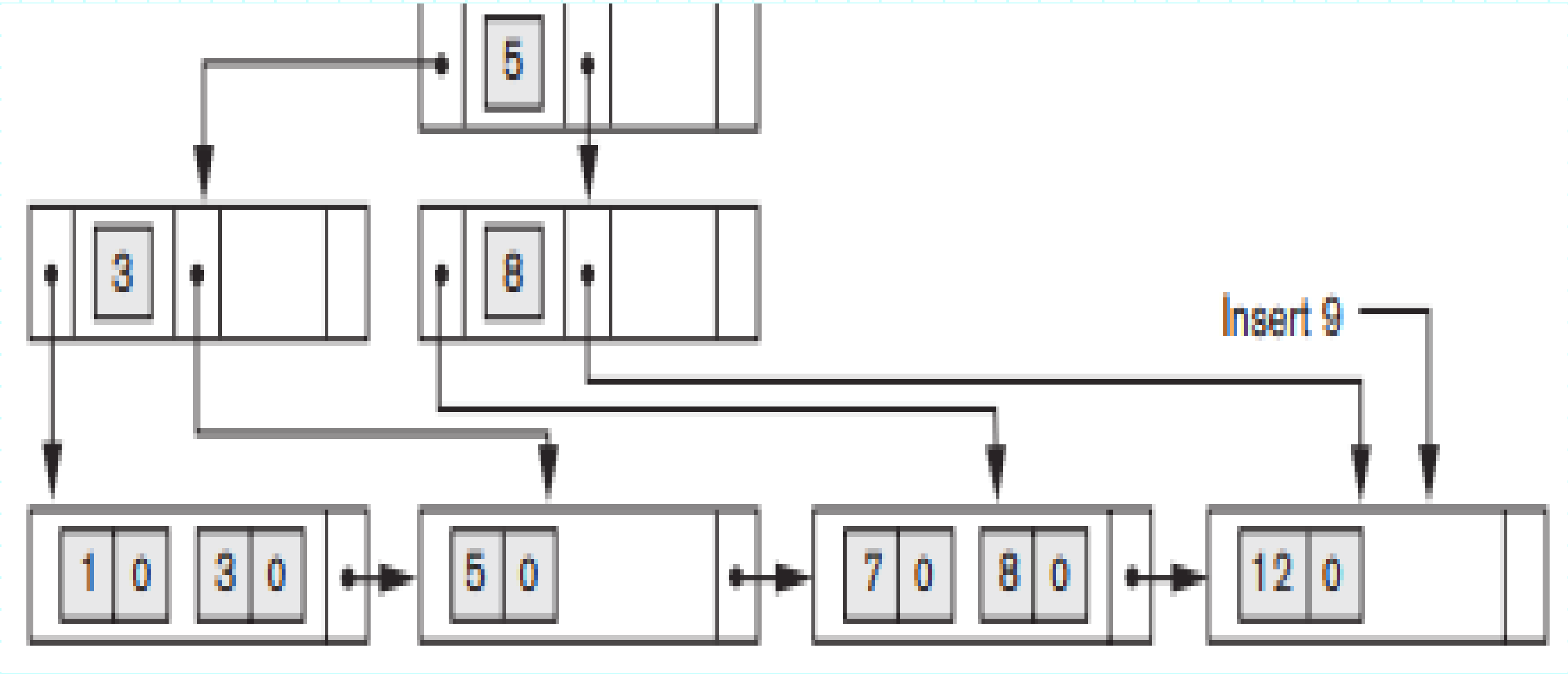
Insertion

Insert a 12:



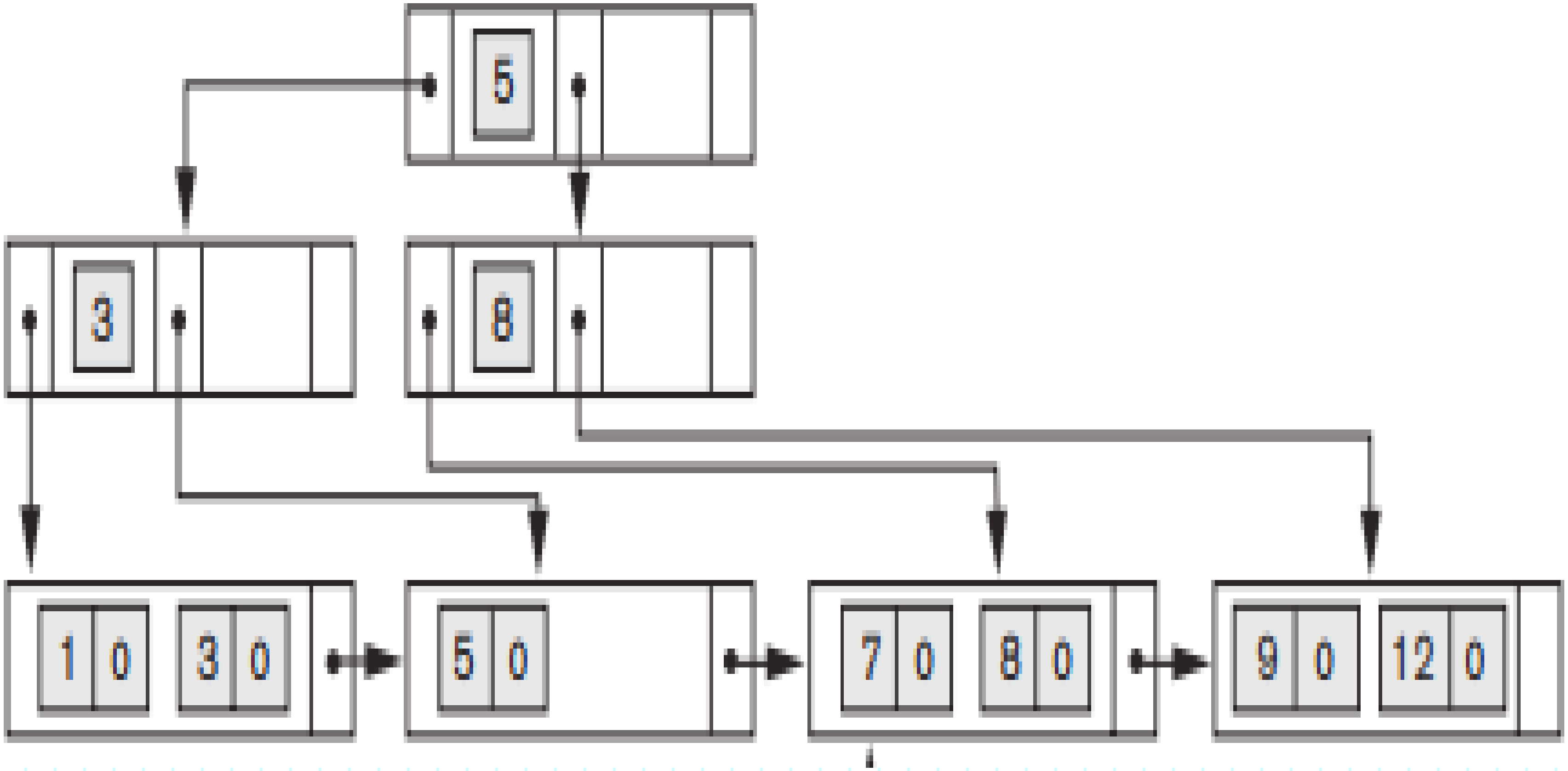
Insertion

Insert a 9:



Insertion

Insert a 6:



Deletion

Search for the node where the new record should go
If the node is more than half full, remove entry and done
Else:

  If sibling (with same parent) is more than half full, take an entry from it, update parent

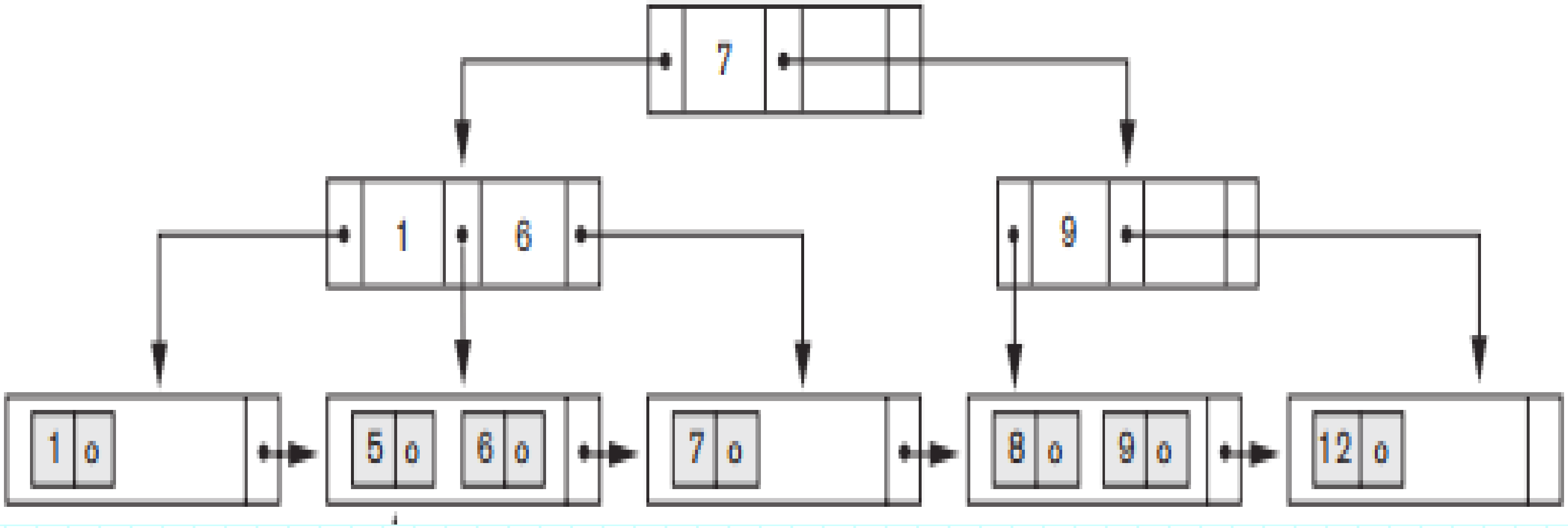
  Otherwise

  - Merge node with sibling
  - Delete entry from parent of removed node
  - May cause parents to merge

– Merging internal nodes:
  - Grab value from sibling and “push through”
  - Update pointers

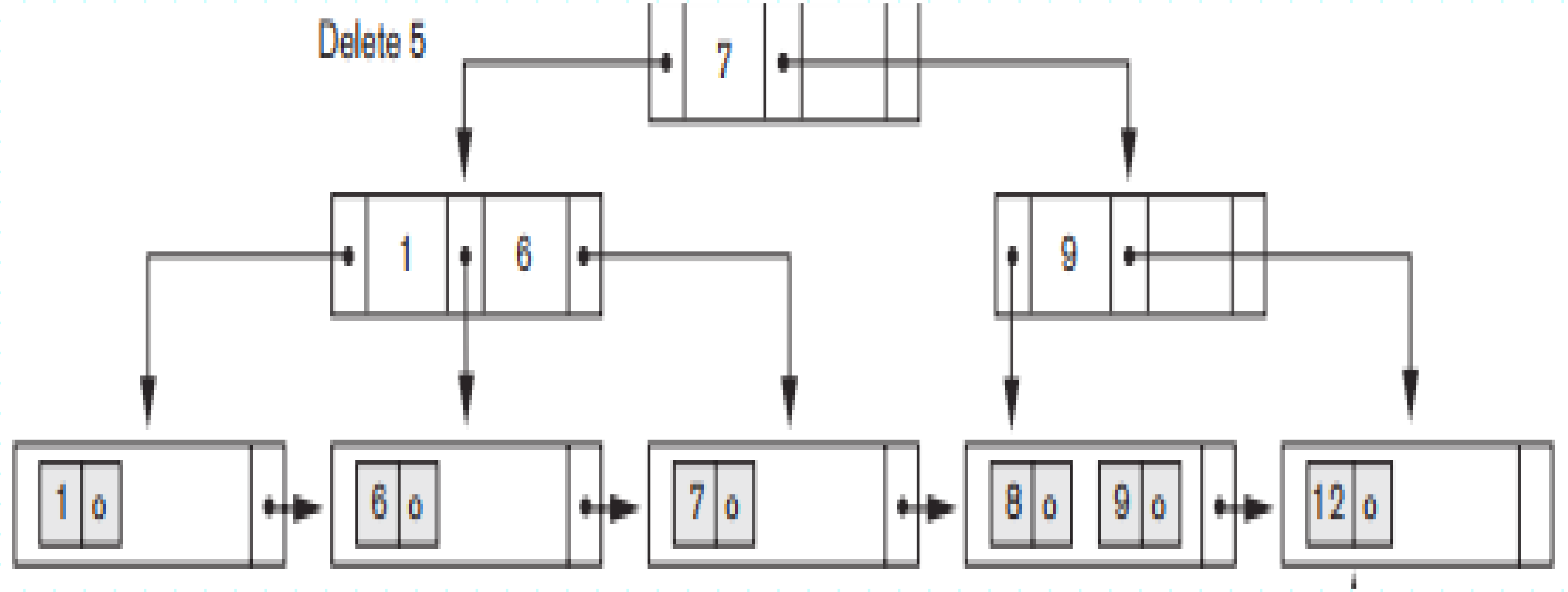
Deletion

Delete 5:



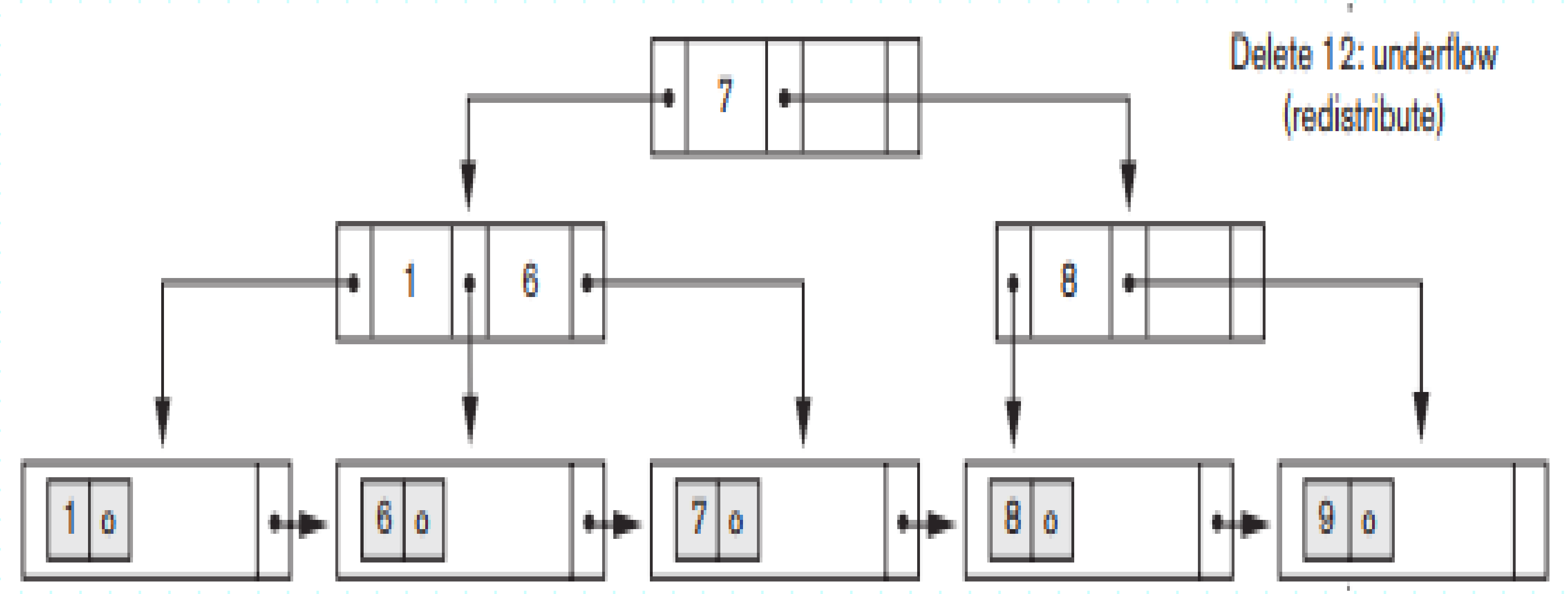
Deletion

Delete 12:



Deletion

Delete 9:



Delete 12: underflow (redistribute)

Exercise

Using a btree with $p_{inner} = 3$ and $p_{leaf} = 2$, sketch the result of the insertion of the following values:

9, 4, 12, 7, 2, 6, 1, 3, 10

Repeat the above example with a btree with $p_{inner} = 4$ and $p_{leaf} = 3$

Exercise

Using the btree (p_{inner} = 3 and p_{leaf} = 2)from the previous exercise, show the result of deleting the following values:

7, 3, 4, 10, 2

Repeat the above example with your p_{inner} = 4 and p_{leaf} = 3 tree

Indexing in SQL

- What columns should we be indexing?
  - Should we just index all columns?

- What part of the query process is affected by the index?

SQL Example

- CREATE INDEX name_idx on Student (lname);

- Can include multiple columns as part of an index
  - How does this affect the index structure?

Exercise

- Revisit your course tracking database. Decide what columns would be worthy of an index, and create them.