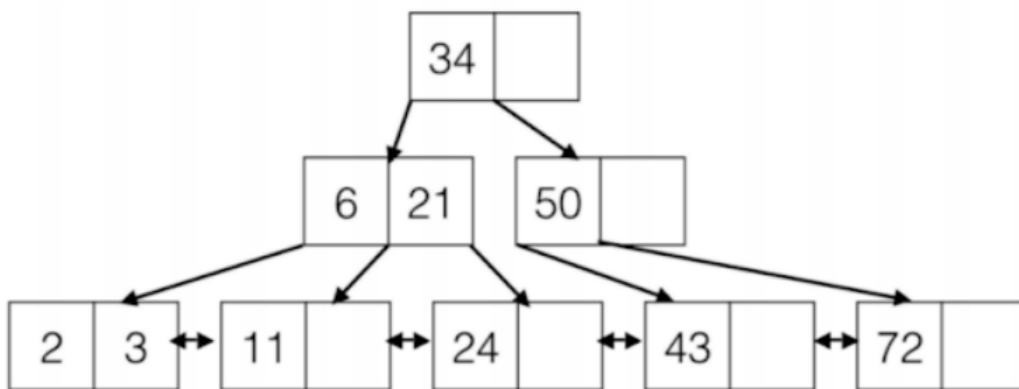


1 Indices (B+ Trees)

Assume we have the following B+ Tree of order 1. Each index node must have either 1 or 2 keys (2 or 3 pointers), and the leaf nodes can hold up to 2 entries.



- (a) What is the maximum number of insertions we can do without changing the height of the tree?
- (b) What is the minimum number of keys you could insert to change the height of the tree?

2 Indices

Two sets of terminology:

Clustered vs. unclustered

- In a clustered index, the data pages are sorted using the same index that build the B+ Tree. This means the keys are roughly sorted in the same order as the data pages, so it'd cost ~1 I/O per page of records wanted.
- In an unclustered index, the data pages are unsorted, meaning chances are it'd cost ~1 I/O per record wanted.

Three alternatives for storing underlying data:

- Alternative 1 (by value): the entire record is directly stored in the leaf page
- Alternative 2 (by reference): (key: ptr to rid) pairs are stored in the leaf pages, where each key may not be distinct
- Alternative 3 (by reference): (key: [ptr to rid1, ptr to rid2, ...]) pairs are stored in the leaf pages, where each key is distinct

(a) Is it possible to have two clustered indices on separate columns?

Suppose we have an alternative 2 unclustered index on (assignment_id, student_id) with a height of 3 (one must traverse 3 index pages to reach any leaf page).

Here is the schema:

```
CREATE TABLE Submissions (  
    record_id integer UNIQUE,  
    assignment_id integer,  
    student_id integer,  
    time_submitted integer,  
    grade_received byte,  
    comment text,  
    regrade_request text,  
    PRIMARY KEY (assignment_id, student_id));  
  
CREATE INDEX SubmissionLookupIndex ON Submissions (  
    assignment_id, student_id);
```

Assume the table and its associated data takes up 12 MB on disk (1 MB = 1024 KB) and that page size is 64 KB. (This includes extra space allocated for future insertions.)

(b) We want to scan all the records in Submissions. How many I/Os will this operation take?

(c) UPDATE Students SET grade_received=85 WHERE assignment_id=20 AND student_id=12345; How many I/Os will this operation take?

(d) In the worst case, how many I/Os does it take to perform an equality search on grade_received?

3 Bulk-Loading

Suppose we were to create an order $d=2$ B+ tree via bulk-loading with a fill factor of $3/4$. Here, fill factor specifies the fill factor for leaves only; inner nodes should be filled up to full and split in half exactly.

We insert keys with all integer values from 1-16 in order. Draw out the final B+ tree. What is its height?