CS186 Discussion Section Week 3 Solutions
Tree-Structured Indexing
Fall 2013

1. Why do we use tree-structured indexes?
To speed up selection (lookups and range) on search key fields. Can have different indexes on different search keys. A file can only be sorted according to one key.

2. What is the difference between an ISAM and B+ Tree Index?

- ISAM Tree: Static structure. Consists of root, primary leaf pages and overflow pages. Long overflow chains can develop.
- B+ Tree: Dynamic structure. Height balanced. Usually preferable to ISAM.
    - Order *d*: Each node contains $d \le m \le 2d$ entries
    - Height: Length of Path from the root to a leaf node
    - Fanout of a node: The number of pointers out of the node

3. We are using a B+ tree with alternative 1 (actual data records in leaf pages) to store one billion records. Each records is 200 bytes, each disk page has 16kB (16,384 Bytes) and will always be at most 67% full.

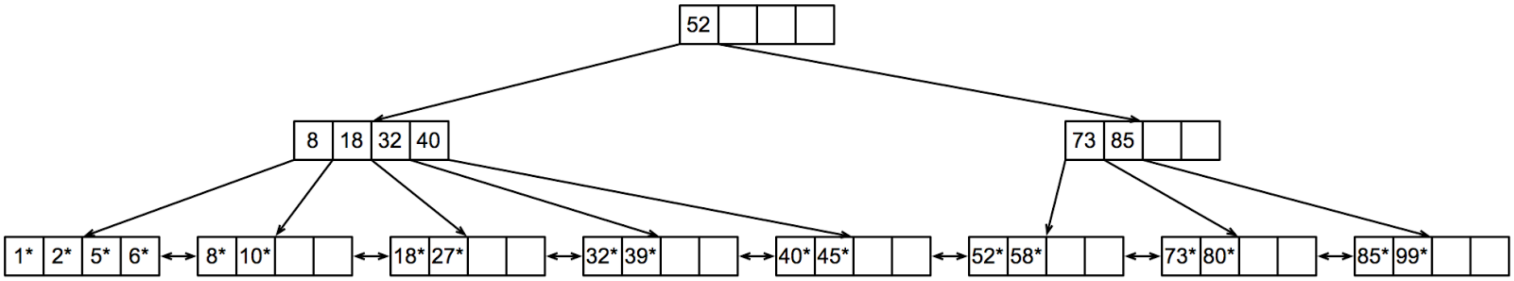
1. How many leaf pages are required?
    - **16384 * 0.67 / 200 = ~54 Entries per page. 10^9 / 54 = ~18.5 * 10^6 pages.**
2. Assume each index entry takes 32 bytes. What is the maximum fanout of the index?
    - **16384 * 0.67 / (32) = ~343**
3. What is the height (# levels of non-leaf nodes) of the tree? How many I/O operations are required to insert a new record (assuming there is enough space in the leaf page)?
    - **Height = log_343(18.5 * 10^6) = ~3.**
      We need 4 Reads (3 non-leaf reads, 1 leaf read) + 1 Write = 5 I/O's.
4. How many pages are required to store the non-leaf nodes?
    - **1 + 343 + 343^2 = 117993**

4. You have decided to develop a new deals website CalDeals which pushes nearby deals to user's mobile phones based on their age group. As you are expanding you realize that your service is getting slower, probably a result of the 2 million users in your database. Assume that each user entry is 2kB in size and that you are mainly performing range queries based on a user's age. Assume the page size is 16kB. Answer the following questions:

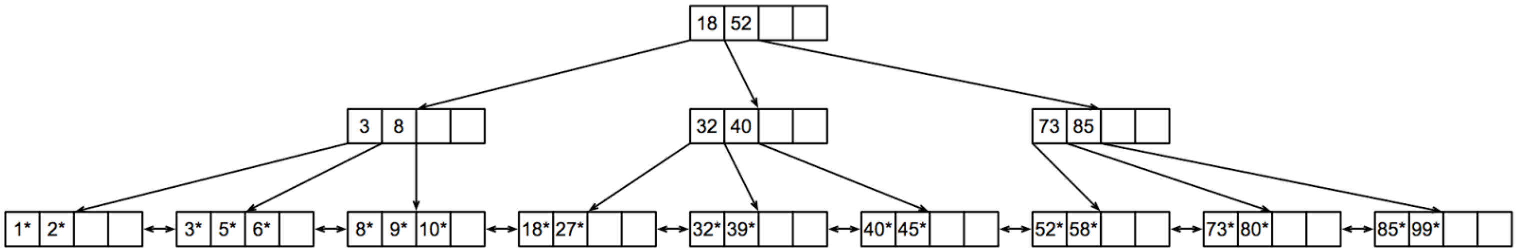
1. You are storing all your data in a heap file. In the worst case, how many I/O operations are necessary to find all users in a certain age range?
    - **Need to scan the whole file. 2 * 10^6 / (16/2) = 250,000 I/O's. (Sequential Access)**
2. You have decided to create a clustered B+-Tree on the age field. The tree has a fanout of 200 and a height of 3. Assume that you are on average returning 50,000 users per query. On average, how many I/O's are perfomed by such a query?
    - **3 + (50,000 / (16/2)) = 6,253**
3. Assume your B+ tree is unclustered. In the worst case, how many I/O's do you need now?
    - **3 I/Os to descend non-leaf index pages**
    - **Assuming that index entries are 3 times smaller than full records,**
      ceil(50,000 / (16/⅔) = 2084 I/Os to read data entries (leaf index pages)
    - **50,000 I/Os to read unordered data pages.**
    - **So 3 + 2084 + 50,000 = 52,087 I/Os.**

5. Consider the B+ Tree below and perform the following operations in order (split full leaf nodes):

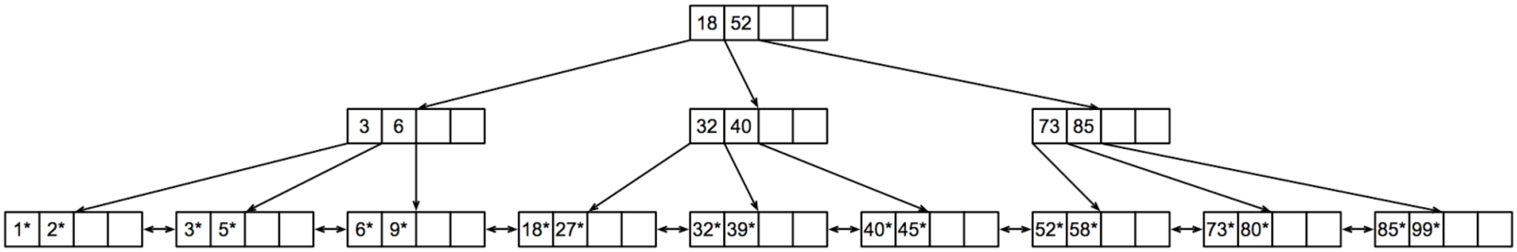
1. Insert 9 and 3.
2. Delete 8 and 10.
3. Insert 46 and delete 52.



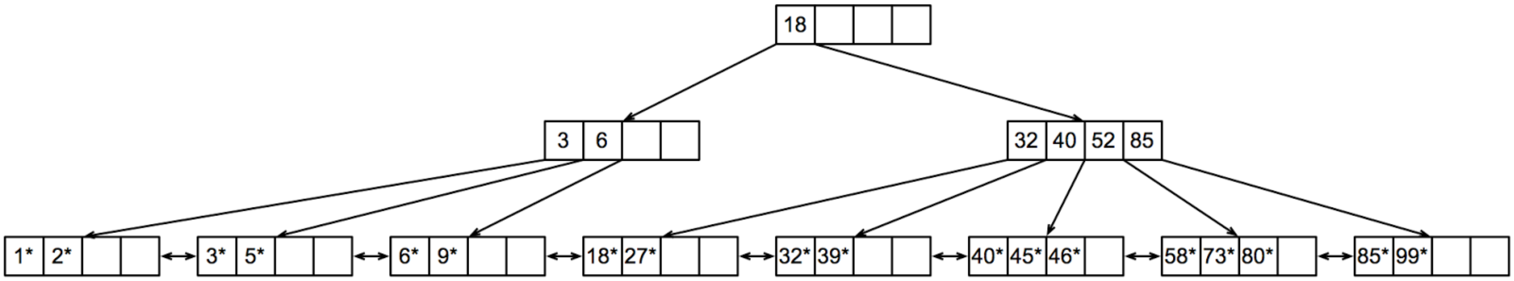
insert 9 and 3



delete 8 and 10



insert 46 and delete 52


18
3 6
32 40 52 85
1* 2*
3* 5*
6* 9*
18* 27*
32* 39*
40* 45* 46*
58* 73* 80*
85* 99*