CS186 Discussion Section Week 3
Tree-structured Indexing
Fall 2013

1. Why do we use tree-structured indexes?

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

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?

    2. Assume each index entry takes 32 bytes. What is the maximum fanout of the index?

    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)?

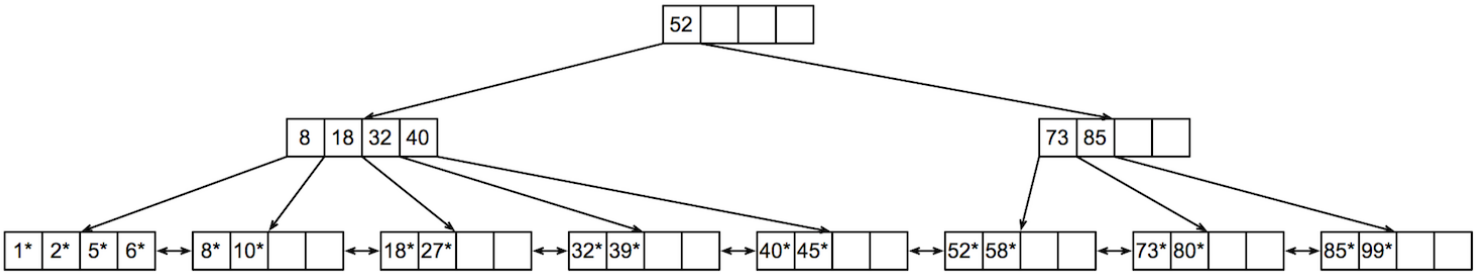
    4. How many pages are required to store the non-leaf nodes?

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?

    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 performed by such a query?

    3. Assume your B+ tree is unclustered. In the worst case, how many I/O's do you need now?

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.


52
8
18
32
40
73
85
1*
2*
5*
6*
8*
10*
18*
27*
32*
39*
40*
45*
52*
58*
73*
80*
85*
99*