Today's announcements:

MP5 available, due 3/29, 11:59p.

Exam 2: 4/2, 7-9p, locations on website.

Class cancelled 4/1.

Exam reviews: 4/1, 12-2p in Siebel 1404

4/1, 8-10p in Siebel 0216

MP5 Solution Party: TBA

Exam 2: 5 questions

1. MC
2. Running times
3. MP4ish
4. MP5ish
5. ???

AVL tree analysis:

Putting an upper bound on the height for a tree of n nodes is the same as putting a lower bound on the number of nodes in a tree of height h.

- Define N(h):

$N(h)$ = the least number of nodes in a tree of height h.

- Find a recurrence for N(h):

$N(h) = 1 + N(h-1) + N(h-2),\ N(0)=1,\ N(1)=2$

- Simplify the recurrence:

$N(h) \quad 2N(h-2),\ N(0)=1,\ N(1)=2$

- Solve the recurrence: (guess a closed form)

$N(h) \quad 2^{h/2},\ h \quad 0$

AVL tree analysis:

- Thm: An AVL tree of height h has at least $2^{h/2}$ nodes, ________.

Consider an arbitrary AVL tree, and let h denote its height.

Case 1: ________

Case 2: ________

Case 3: ________ then, by an Inductive Hypothesis that says

________________________________, and since

________________________________, we know that

________________________________.

Punchline:

Classic balanced BST structures:

- Red-Black trees – max ht $2\log_2 n$.

  Constant # of rotations for insert, remove, find.

- AVL trees – max ht $1.44\log_2 n$.

  O(log n) rotations upon remove.

Balanced BSTs, pros and cons:

- Pros:
  - Insert, Remove, and Find are always O(log n)
  - An improvement over:
  - Range finding & nearest neighbor
- Cons:
  - Possible to search for single keys faster
  - If data is so big that it doesn't fit in memory it must be stored on disk and we require a different structure.

B-trees (the only “out of core” data structure we’ll discuss)

Can we always fit data into main memory?

So where do we keep the data?

Big-O analysis assumes uniform time for all operations.

But…

The Story on Disks

2GHz machine gives around 2m instructions per _____.

Seek time around ________ for a current hard disk.

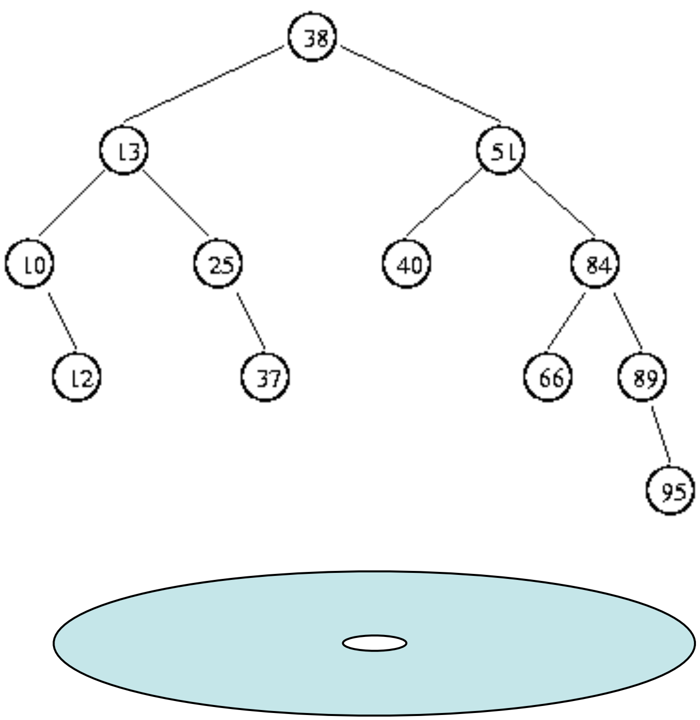
Imagine an AVL tree storing US driving records.

- How many records?
- How much data, in total?
- How deep is the AVL tree?

- How many disk seeks to find a record?

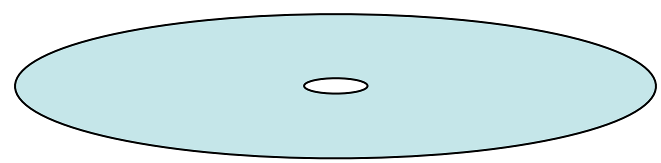
B Trees

Suppose we weren't careful…



B Tree of order m

12	18	27	52	58	63	77	89



1. relevant data
2. shallow tree

B-tree Goals

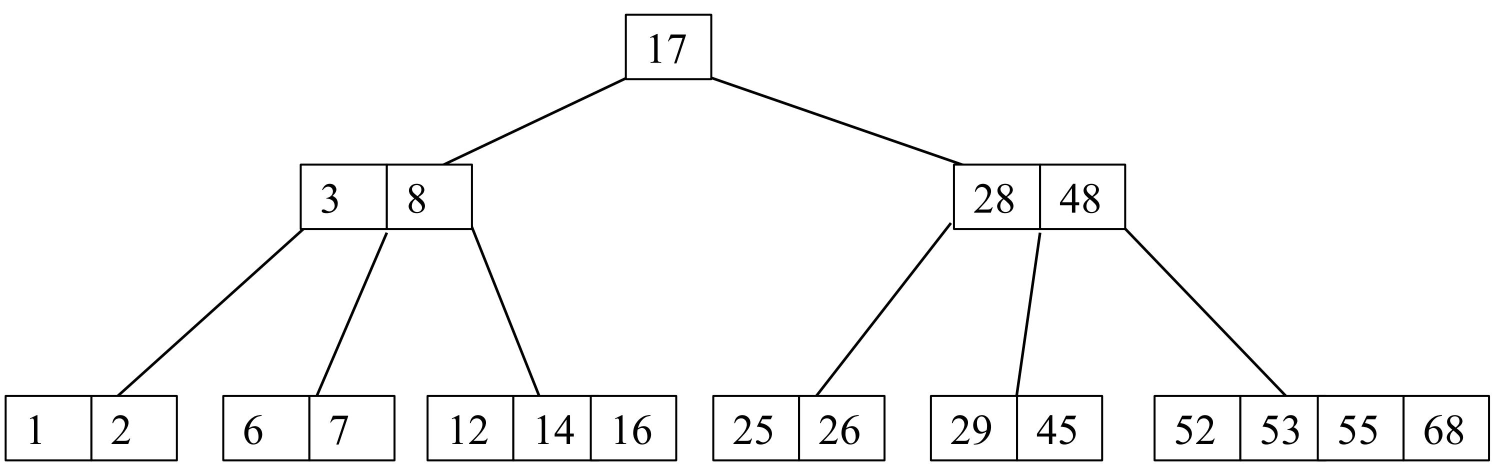
- Minimize the number of reads from disk
- Build a tree that uses 1 disk block per node
  - Disk block is the fundamental unit of transfer
- Nodes will have more than 1 key
- Tree should be balanced and shallow
  - In practice branching factors over 1000 often used

http://people.ksp.sk/~kuko/bak/big/

Definition of a B-tree

B-tree of order *m* is an *m*-way tree

- For an internal node, # keys = #children -1
- All leaves are on the same level
- All leaves hold no more than *m*-1 keys
- All non-root internal nodes have between $\lceil m/2 \rceil$ and m children
- Root can be a leaf or have between 2 and *m* children.
- Keys in a node are ordered.



Searching a B-tree

```
bool B-TREE-SEARCH(BtreeNode & x, T key){
   int i = 0;
   while ((i < x.numkeys) && (key > x.key[i]))
      i++;
   if ((i < x.numkeys) && (key == x.key[i]))
      return true;
   if (x.leaf == true)
      return false;
   else{
      BtreeNode b=DISK-READ(x.child[i]);
      return B-TREE-SEARCH(b,key);
   }
}
```