For Friday

- Read Weiss, chapter 5, sections 1-4
- Paper 1 due

Paper 1

- Any questions?

Splay Trees

- Interested in the cost of a sequence of search operations rather than the cost of a single search.
- We want to make sure that the **amortized** cost of M search operations is M log N.

Basic Idea

- When we find a node, we're going to rotate it to the top in a way that helps to balance the tree if it is currently unbalanced.

Cases

- Found node has no grandparent: rotate node and root
- Found node has a grandparent:
  - zig-zig case (parent is same side of grandparent that node is of root): rotate node and grandparent
  - zig-zag case (node's value is in-between value of parent and grandparent): do a standard AVL double rotation

Comparison

- Splay trees and AVL trees

External Dictionaries

- We've talked so far about dictionaries small enough to reside in memory
- However, many applications require dictionaries much larger than will easily fit in memory
- Biggest issue for external dictionaries is the number of disk accesses required for an operation
- Each disk access retrieves a block of memory

m-way Search Trees

- Empty tree
  or
- Each internal node has up to m children and between 1 and m-1 elements
- A node with p elements has exactly p+1 children
- Elements are ordered

B-tree of Order m

- an m-way search tree
- If non-empty
  - The root has at least two children
  - All internal nodes other than the root have at least ceiling of m/2 children.
  - All external nodes are at the same level
- Thus we have guarantees on the height of the tree
- Book technically covers structure called B+-tree—items in internal nodes also appear in external nodes

Operations

- Searching
- Insertion
- Deletion