Slide 1

Administrivia

- Homework 7 writeup on Web. Design due Thursday, code next Tuesday.

Slide 2

Sorted Binary Trees, Continued

- Key property — everything in the left subtree is smaller than the root, and everything in the right is bigger.
- Last time we did code for `add` and `find`. Now, `remove`.

Slide 3

Priority Queues, Revisited

- Several data structures we could use to implement priority queue ADT:
  - Unsorted linked list.
  - Sorted linked list.
  - Sorted binary tree.

  Compare how much work to add/remove if N elements. Can we do better? Maybe!

Slide 4

Heaps

- Heap is another tree-based data structure, with two properties:
  - A node is always “bigger than” both its children.
  - Tree is “complete”.
- For a priority queue, we want to retrieve the “biggest” thing (for game problem, smallest update time). Does this seem useful?
- Note also that we can store a complete binary tree in an array.
- How to insert and remove? Compare running times.

Slide 5

Minute Essay

- Show what a heap looks like after inserting 5, 4, -1, 10, 6, 20. (Okay to draw tree-based pictures.)