Augmented Data Structures and the Test

10-21-2003

Opening Discussion

- What did we talk about in the class before the test?
- Do you have any questions about the assignment?
- Test Results
  - 5 As, 2 Bs, 4 Cs, 4 Ds, 5 Fs

Test Comments

- Question 1 uses a division hashing scheme with linear probing.
  - Can someone explain % for me?
  - How does linear probing resolve collisions?
  - Show some work so I can give you a bit of partial credit.
- Anything to left of -> (or . in Java) can cause a seg fault (NullPointerException). You need to check. No loops.

More Test Comments

- Smart test taking is just as important as anything. Leaving blanks made my grading easier, but I don't like writing the -10.
- Discussion on the difficulty of the course and the test.

Skip Lists

- This is another data structure that can give you O(log n) behavior. It is not a tree.
- In this data structure we keep multiple parallel sorted linked lists with each list being longer than the one before it. The trick is keeping track of the right elements.

Augmented Data Structures

- This is an idea that we have already talked about, but now we can make it more formal. Sometimes you can use just a basic, preexisting data structure, but often you can't.
- We have already talked about order statistic trees where we put a size in the element. This can be done with a regular tree, or a balanced tree like an AVL or Red-Black tree.

Augmenting and Speed

- The trick to augmenting a data structure is making sure that you can keep the values up to date without messing up the order of the different methods.
- For a tree we can prove augmentation is safe if the value in a node can be calculated from the value in the children.
- This is the case for size and height.

Interval Trees

- Another potential augmentation for a tree is to make an interval tree. This is a tree that keeps tracks of intervals of values instead of single values.
- We sort them by the low end of the interval. To help us use them we also store the maximum of any interval under a given node.
- Again, this works with balanced or unbalanced trees.

Code

- Let's continue writing our binary tree code and now we will augment it so that we have an ordered statistic tree.

Minute Essay

- Draw a picture of what a skip list might look like for the values 1, 3, 4, 5, 7, 9 13, 15.
- Assignment #3 is due a week from today.