CS280 Notes 2/10

by Christian Sagel

You need the Dr.Memory command line calls and the -g command line switch for compiling to get good debugging info.
If there's a memory allocation failure, don't do anything special. Still handle it and wrap in a try catch. Abort when it fails.

The midterm will be 2 weeks from today on February 24.

The monday after next week he will hand out the first assignment.

Implementing Tree Algorithms

Balancing a tree is like sorting a linked list in terms of complexity.
Our trees will be templatized.
We will probably be using an ObjectAllocator for the next assignment.

We want a function call to build the list (tree).
Recursively we would build the head, then build the tail.
Harder to get the recursive version wrong.

Way too
Build the node, build the left tree, the right tree. (Pre-order)
We just have to pick what order want to.

Almost all our functions will be done recursively, only a few lines of code.
1 node is a height of 0.

More tree algorithms: To find the number of nodes in a tree, the height of a tree.
We will be using references to pointers for later…

What's the complexity of counting the nodes in the tree: O(n)
Finding the height of the tree. How many nodes to touch? O(n)
Traverse is also O(n)

Level-order traversal: Traversing all nodes on each level from left to right.
Recursion is not a good way to go for a breadth-depth search.
In recursion the data structure is a stack. Not very good for this.
Instead we want to use a queue.

Level-order traversal touches each node once. An STL queue or a circular array gives us constant push and popback, making it quite efficient.

The directory structure in our OS uses trees. Traversing the file system is done in a tree fashion. Dragging a folder onto another is done recursively by the OS.

Binary Search Trees

A BST is a binary tree in which all the nodes on the left subtree are less than those of the right subtree. There will be no duplicates, just less.

Some operations:

- InsertItem
- DeleteItem
- ItemExists
- Traverse (Pre,In,Post)
- Count
- Height

Notes:
In most of the containers for the rest of the semester, the delete function will be the hardest.

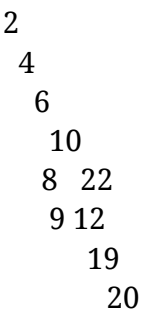
ItemExists and InsertItem are done recursively, with
If find doesn't find an item, it returns where it belongs.
Inserting is easy because once you always insert at the bottom of the tree level.
Because you can delete nodes with children, it can get tricky.

Two sets of numbers:
12,22,8,19,10,19,20,4,2,6

```
      12
   8      22
 4  10  19
2 6 9     20
```

The height of the the tree is 3.
The offending node is 22.

2,4,6,10,8,22,12,9,19,20



Leaf nodes are balanced. Height is 7.
9,19,20.

BSTs are are very sensitive to the order of the data. If you give it unsorted data, it just degenerates into a linked list. The height of the tree is the worst case for finding something.

Hashing randomizes the data.
If it's not balanced, it cannot possibly be complete.
Deleting nodes will require some thought for different cases.
Deleting a leaf, set parent to null.

The smallest value is to recursively look for the leftmost node.
We are always going to replace the root with the predecessor in in-order traversal.
The predecessor is the rightmost node in the left tree.
The worst case si 30 cases.

Mead has a lot of tree code.
Rotating nodes is a fundamental technique that is performed on BSTs.