-A2- 8 days (approx), extra office hours next week
- E6 up this aft (last exercise...).

CSC148 fall 2013

binary search tree

week 8

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/148/F13/
416-978-5899

October 29, 2013

Outline

performance

binary search tree

big-oh

performance...

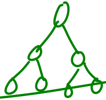
We want to measure **algorithm** performance, independent of hardware, programming language, random events

elements in a list, e.g.

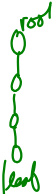
Focus on the **size** of the input, call it n. How does this affect the resources (e.g. processor time) required for the output? If the relationship is linear, our algorithm's complexity is $\mathcal{O}(n)$ — roughy proportional to the input size n.

list searching

$O(\lg n)$.



$\frac{n}{2^k} \le 1$



$n \le 2^k$

$\lg n \le k$

list size n

$\frac{n}{2^k} \le 1$

You've already seen algorithms for seeing whether an element is contained in a `list`:

`[97, 36, 48, 73, 156, 947, 56, 236]`

$O(\lg n)$ → binary search,

What is the performance of these algorithms in terms of list size? What about the analogous algorithm for a tree?

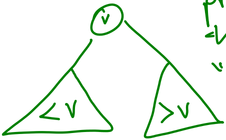
→ linear search. $O(n)$

if list is already sorted; try binary search.

a more efficient binary tree

We need to impose a sorting condition on binary trees. A **binary search tree** is:

- a binary tree ✓



provided tree is "balanced"

- left subtree of every node contains only values smaller than those of that node

- right subtree of every node contains only values greater than those of that node

efficiency?

Binary search of a list allowed us to ignore (roughly) half the list. Searching a binary search tree allows us to ignore the left or right subtree.

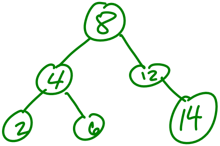
→ AVL trees, red-black trees.

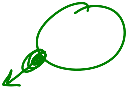
If we're searching the tree rooted at node n for value v, then one of three situations are possible:

- node n has value v
- v is less than node n's value, so we should search to the left
- v is more than node n's value, so we should search to the right

ignore entire subtrees.

insert

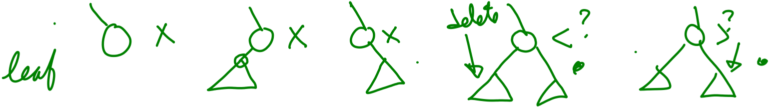




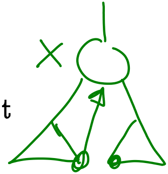
Inserting is closely related to finding a node:

- if we find a node in our tree, no need to insert it!
- otherwise, we find the spot it should be, and insert it there.

deleting



deleting is a bit trickier, because there are several scenarios to consider, even after we've figured out which node we wish to delete:



- if the node we wish to delete is a leaf, just delete it
- if the node we wish to delete has exactly one child, replace it with the other
- if the node we wish to delete has two children, replace it with the largest child in its left subtree...

You should draw some diagrams until you understand these scenarios

deleting

deleting is a bit trickier, because there are several scenarios to consider, even after we've figured out which node we wish to delete:

- if the node we wish to delete is a leaf, just delete it
- if the node we wish to delete has exactly one child, replace it with the other
- if the node we wish to delete has two children, replace it with the largest child in its left subtree...

You should draw some diagrams until you understand these scenarios

deleting

deleting is a bit trickier, because there are several scenarios to consider, even after we've figured out which node we wish to delete:

- if the node we wish to delete is a leaf, just delete it
- if the node we wish to delete has exactly one child, replace it with the other
- if the node we wish to delete has two children, replace it with the largest child in its left subtree...

You should draw some diagrams until you understand these scenarios

deleting

deleting is a bit trickier, because there are several scenarios to consider, even after we've figured out which node we wish to delete:

- if the node we wish to delete is a leaf, just delete it
- if the node we wish to delete has exactly one child, replace it with the other
- if the node we wish to delete has two children, replace it with the largest child in its left subtree...

You should draw some diagrams until you understand these scenarios