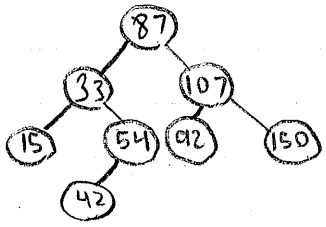
Algo

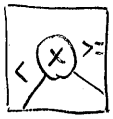
Efficiency : Trees

9/28

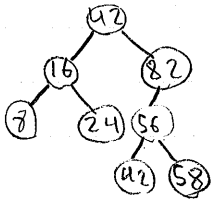
What is binary search tree?



add 42



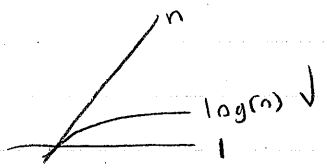
42, 16, 24, 82, 56, 58, 42, 8



$n = 8$

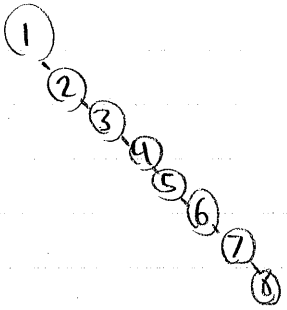
What are your bc and wcase scenarios?

wctime = 4 wc < n



What if inserting in-order list?

1 – 8



In-order traversals give back "in order"

Problems with Binary search tree

- Imbalance makes worst-case O(n)

Delete?

- Just remove leaf — 0 children
- Shift up and delete middle node — 1 child
- Go right, then as far left as possible, swap values, then easy delete — 2 children

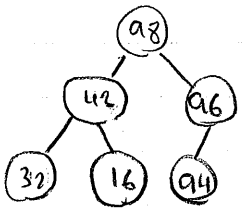
Algo

Heap (Datastructure)

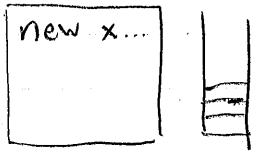
9/28

heap vs. heap

Datastructure



w/Stack



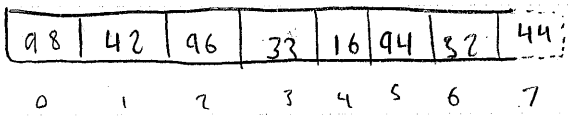
★Children smaller than parent

How to insert new largest value?

1. Put in next complete spot
2. Check "less than parent" and swap child w/parent until true

3 big ideas for heap

1. Insert
2. Find max (always root)
3. Delete



Formulas for heaps

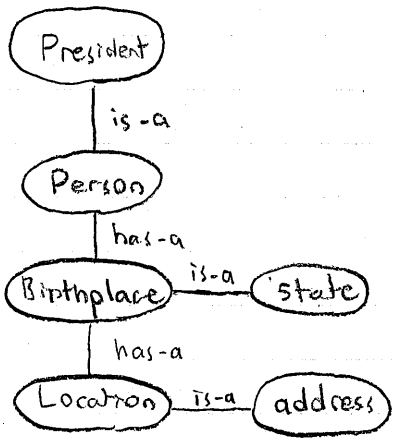
$$\text{parent}(n) = \lfloor (n-1)/2 \rfloor, \quad n \neq 0$$

$$\text{LChild}(n) = 2n+1 \quad \text{if } 2n+1 < \text{size}$$

$$\text{RC}(n) = 2n+2 \quad \text{" } 2n+2 < \text{size}$$

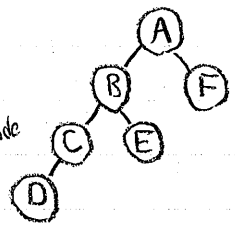
Algo

What are is-a & has-a relationships?



Properties of trees

→ Can't have two parents (no cycles)



Level
0
1
2
3

Height = 4

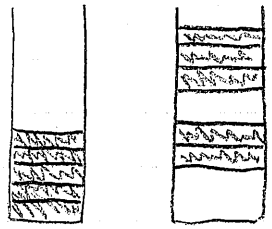
- Root • First node; no parent; can be leaf or internal node
- Node • Node; has value, children
- Edge • "Branch" connecting parent & child
- Parent • Node before selected node
- Child • Node after selected node
- Sibling • Node on same level as selected
- Ancestor • Anyone higher up
- Descendent • Anyone lower down path
- Subtree • A section that could stand as tree by self
- Path • e.g. path A → E is ABE
- Depth • How many edges
- Height • Deepest depth + 1; Max nodes down
- Level • Begins at 0 and is not consistently termed
- Internal tree • has children
- Leaf • no children
- * n-ary • any amnt of children

	Data	LC	RC
0	A	1	5
1	B	2	4
2	C	3	-1
3	D	-1	-1
4	E	-1	-1
5	F	-1	-1

Algo

Stack vs Heap

Overview



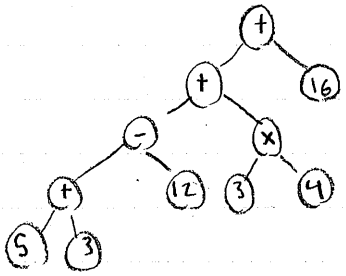
9/26 - 2

Note on pointers vs array

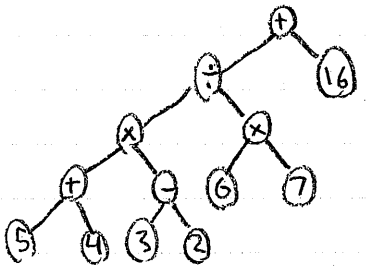
Pointers being read from memory in tree takes too long. Having close objects in memory goes faster. (read array)

Fun w/ math

((((5+3) - 12) + (3 x 4)) + 16)



((((5+4) x (3-2))/ (6 x 7)) + 16)



13 nodes

p	
0	
1	
2	H = 5
3	
4	

What is "complete" vs. "Full"?

Full: all nodes have 0, or 2 children

Complete: Entire level is filled for all levels
- except last, filled in left to right

"Balance"?

Equal data on each "side"

Algo

9/26 - 3

Balanced nodes formula

h	# nodes	level #
1	1	1
2	3	2
3	7	4
		8
		16

$2^h - 1$

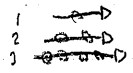
What does this say about complete

Determine min & max

How to traverse binary tree?

Breath - first

Depth - first

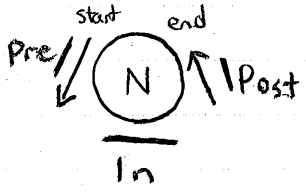
Level - by - level 

Pre - order - Parents first

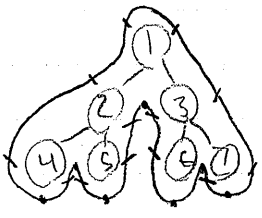
Post - order - Children first

In - order - Left, middle, right  ← Binary Only

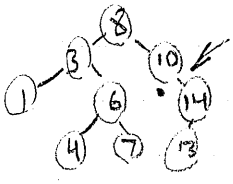
When is something traversed / printed?



eg. pre → 1 2 4 5 3 6 7 \
post → 4 5 2 6 7 3 1 /
in → 4 2 5 1 6 3 7 •



pre 8 3 1 6 4 7 10 14 13
post 1 4 7 6 3 13 14 10 8
in 1 3 4 6 7 8 10 13 14
lev 8 3 10 1 6 14 4 7 13



note the underside for in-order