DATA STRUCTURE OPERATIONS (BIG O) (AVG, WORST)

	ACCESS	SEARCH	INSERTION/DELETION
ARRAY	1, 1	N, N	N, N
STACK	N, N	N, N	1, 1
LINKED LIST	N, N	N, N	1, 1
HASH TABLE	/	1, N	1, N
BST	$\log N$, N	$\log N$, N	$\log N$, N
LLRB	$\log N$, $\log N$	$\log N$, $\log N$	$\log N$, $\log N$

NOTE: INSERTION SORT RUNS $\Omega(N)$, $O(N^2)$ GENERALLY, NO WORSE THAN # OF INVERSIONS

ARRAY SORTING ALGORITHMS (BIG O)

	BEST	AVERAGE	WORST
QUICKSORT	$N \log N$	$N \log N$	N^2
MERGESORT	$N \log N$	$N \log N$	$N \log N$
HEAPSORT	$N \log N$	$N \log N$	$N \log N$
SELECTION SORT	N^2	N^2	N^2

	HEAPIFY	FINDMAX	REMOVE MAX	INSERT/DELETE	MERGE
HEAP	N	1	$\log N$	$\log N$	M+N

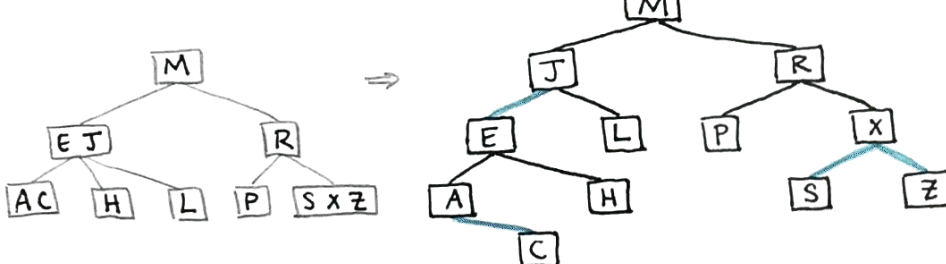NOTE: IF ADDING ONLY LARGE VALUES, INSERTION TAKES CONSTANT TIME

$$O(1) < O(\log N) < O(N) < O(N \log N) < O(N^2) < O(N^2 \log N) < O(N^3) < O(2^n) < O(n!) < O(N^N)$$

2-3 TREE: MAX 2 ELEMENTS/NODE AND 3 CHILDREN   **2-3-4 TREE:** 1, 2, OR 3 ELEMS/NODE AND 2, 3, 4 CHILDREN RESPECTIVELY
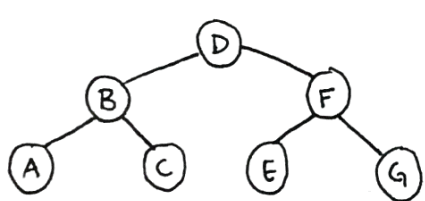
INSERTION INTO JUICY NODE

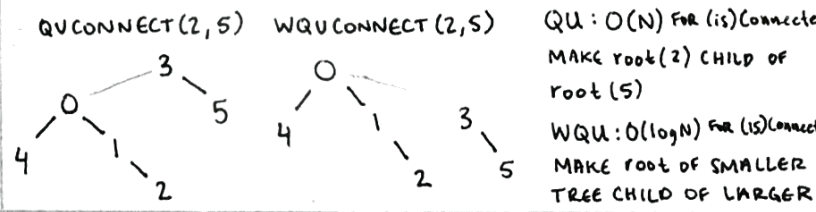INSERT t, u

INSERT y, z
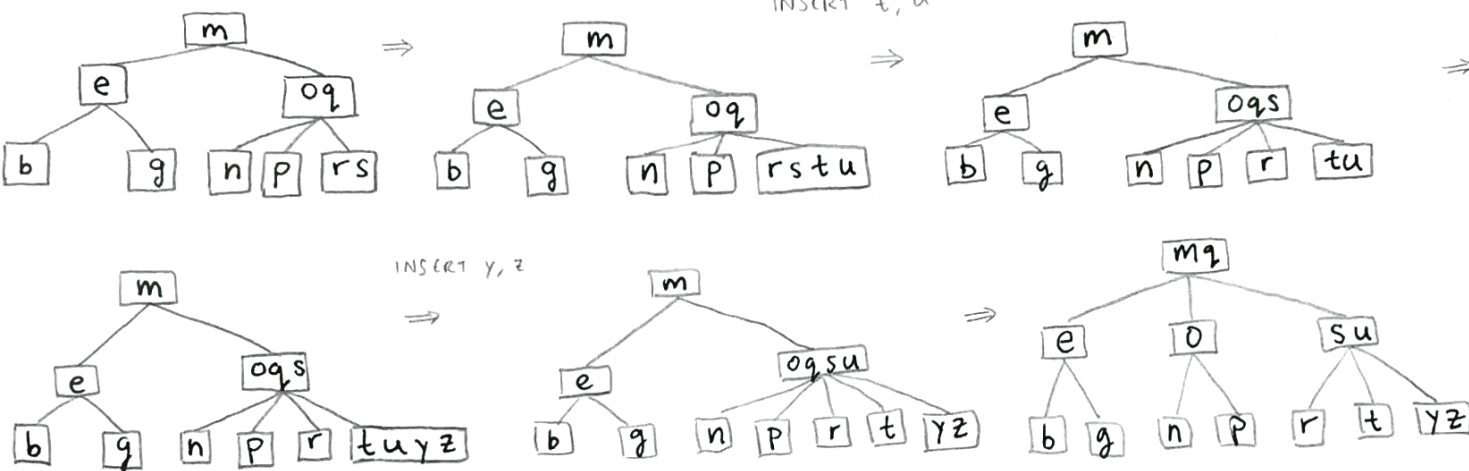
2-3-4 TREE TO LLRB

TREE TRAVERSALS

ORDER VISITED BY BFS FROM ROOT

LEVEL ORDER: D B F A C E G

PREORDER: D B A C F E G

INORDER: A B C D E F G

POSTORDER: A C B E G F D

WEIGHTED QUICK UNION VS. QUICK UNION

QU CONNECT (2, 5)   WQU CONNECT (2, 5)

QU: $O(N)$ FOR (IS) CONNECTED
MAKE root(2) CHILD OF root(5)

WQU: $O(\log N)$ FOR (IS) CONNECTED
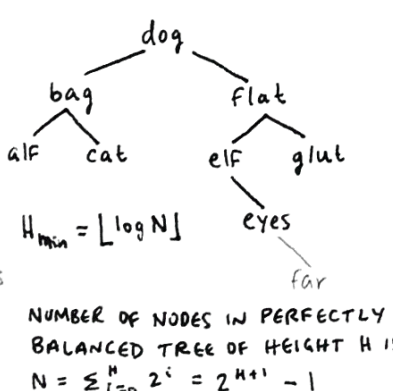MAKE ROOT OF SMALLER TREE CHILD OF LARGER

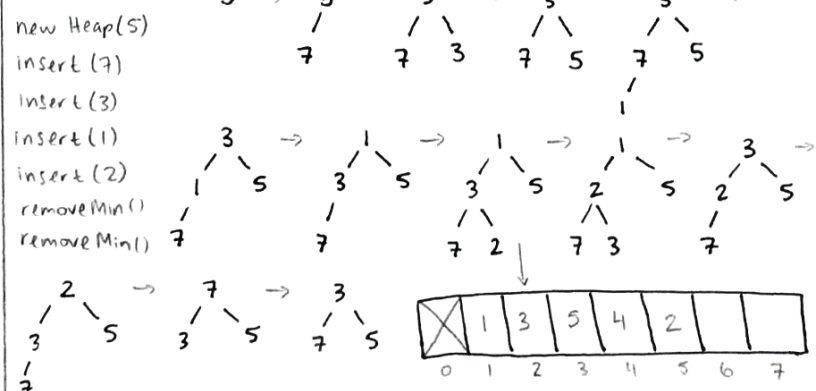## BSTs DELETING

DELETING W/ NO CHILDREN:
JUST SEVER THE LINK

DELETING W/ ONE CHILD:
MAKE CHILD INTO CHILD OF YOUR PARENT

DELETING W/ TWO CHILDREN:
CHOSE ONE OF INNER-MOST NODES WITH 0 OR 1 CHILDREN TO BE THE NEW ROOT WHICH WILL TURN INTO CASE 1 OR 2.

BST INSERTING

$H_{min} = \lfloor \log N \rfloor$

NUMBER OF NODES IN PERFECTLY BALANCED TREE OF HEIGHT H IS
$$N = \sum_{i=0}^{H} 2^i = 2^{H+1} - 1$$

HEAPS

new Heap(5)
insert (7)
insert (3)
insert (1)
insert (2)
removeMin()
removeMin()

TILDE, FINDING a AND b

ASSUME $R(N)$ APPROACHES aN^b

N	R(N)
500	0.08 sec
1000	0.70 sec
2000	1.22 sec
4000	5.15 sec
8000	43.62 sec

FINDING b MATHEMATICALLY

$$\frac{R(8000)}{R(4000)} = \frac{a\,8000^b}{a\,4000^b}$$

$$\frac{43.62}{5.15} = \left(\frac{8000}{4000}\right)^b$$

$$8.47 = 2^b \rightarrow b = 3.08$$

FINDING b BY APPROXIMATION

TAKE THE LARGEST 2 ENTRIES. NOTE THAT WHEN N DOUBLES,

$43.62 \approx 5.15(9)$

$R(N)$ ABOUT $9\times$ BIGGER. THUS, $b \approx 3$.

FINDING a

$$R(N) \approx aN^b$$

$$43.62 = a(8000)^3$$

$$43.62 = a(8\times10^3)^3$$

$$a \approx \frac{43.62}{512}\cdot 10^{-9}$$

$512\overline{)43.62}$ .00.08 r $\frac{226}{512}$
4096
266

SO $a \approx 0.085 \times 10^{-9}$

OR $a \approx 8.5 \times 10^{-11}$

$$R(N) \sim (8.5\times10^{-11})N^3$$

SUMMATIONS

SUM	FORMULA	Θ	$\sim$
$1 + 2 + 3 + \cdots + N$	$N(N+1)/2$	N^2	$\frac{1}{2}N^2$
$1 + 2 + 4 + 8 + \cdots + N$	$2N-1$	N	$2N$
$1 + 3 + 5 + \cdots + N$	N^2	N^2	N^2
$1 + 4 + 9 + 16 + \cdots$	$\dfrac{N(N+1)(2N+1)}{6}$	N^3	$\frac{1}{3}N^3$

HASHMAPS

$$L = \frac{N}{M}$$