COP 3503 1/11/24

DJ HashSet, PriorityQueue, ArrayDeque, Custom Sorting
HashMap
TreeSet
TreeMap

Kattis { bst
summertrip

HashSet: add, remove, search O(1) expected time

CD problem adding each

CD, duplicates weren't counted, so

I/E principle

What's the difference btw Set and map?

Map: each item in Set is mapped to some piece of information

ADAMS → 2
BURNS → 1
ZIN → 2
ELDRICK → 1

ADAMS
ADAMS
BURNS
ZIN
ZIN
ELDRICK

add, remove, search
change mapped value
retrieve mapped value

Common Use of Map

HOUSE → 0
GYM → 1
SCHOOL → 2
etc.
mapping locations to idnum

SAM → 407xxxxxxx
ELLIE → 321xxxx

phone-num

election2.java

TreeSet - ordered

implemented w/a balanced binary search tree.
NO DUPLICATES
"SORTED" (can recreate sorted order efficiently)

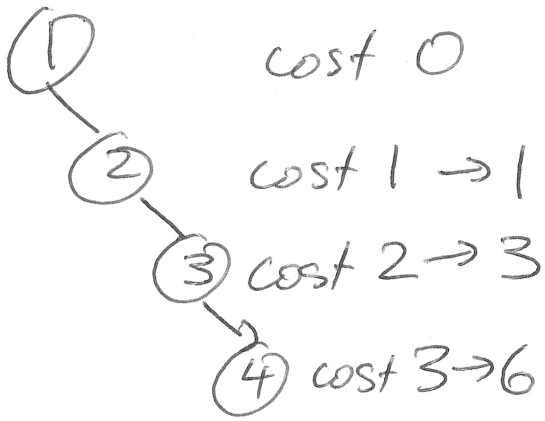
add
remove
search
} O(lgn) n = # items

8 methods

lower(x) - return largest elem set < x
ceiling(x) — least element ≥ x
first — returns smallest
last — returns largest
higher - returns smallest > x
pollFirst → removes return smallest
pollLast → remove return largest

bst katis

inp	out
4	0 ✓
1	1 ✓
2	3 ✓
3	6 ✓
4	



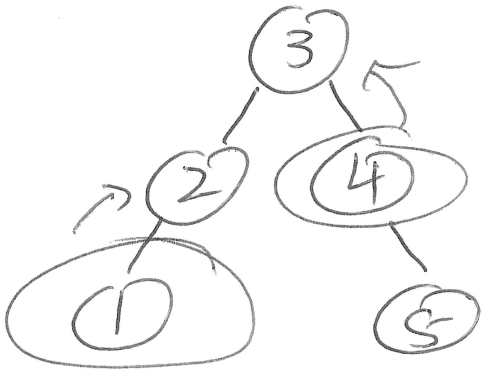
Can we simulate?

NO

worst case: 1, 2, 3, ... > 300000

sum of depth $0+1+2+\cdots+299,995 \approx \frac{300000^2}{2}$ bis!

inp	out
5	
3	0
2	1
4	2
1	4
5	6



0

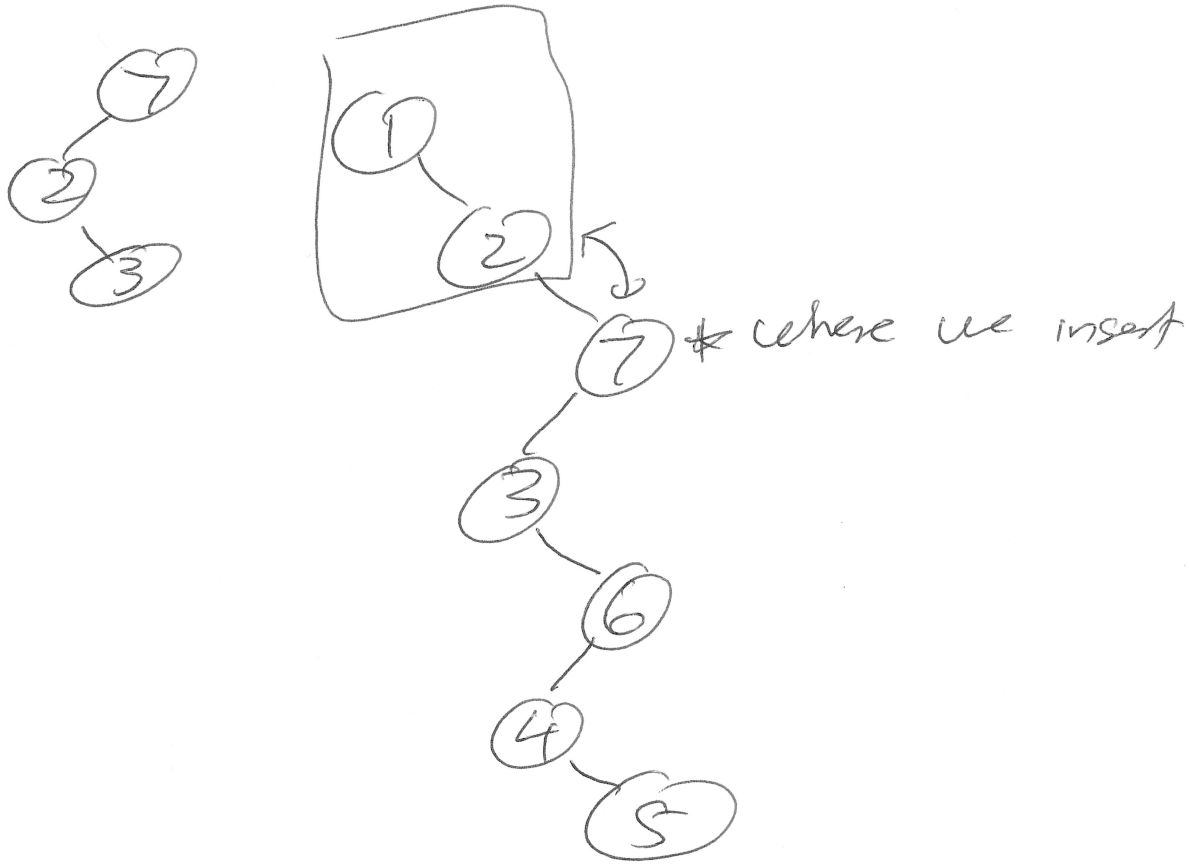
1

1 → 2

2 → 4

2 → 6

1
2
~~7~~ 7
3
6
4
5



When I insert a value into bst, there are ONLY 2 possible candidates for the parent:

lower (a) greatest item less than it

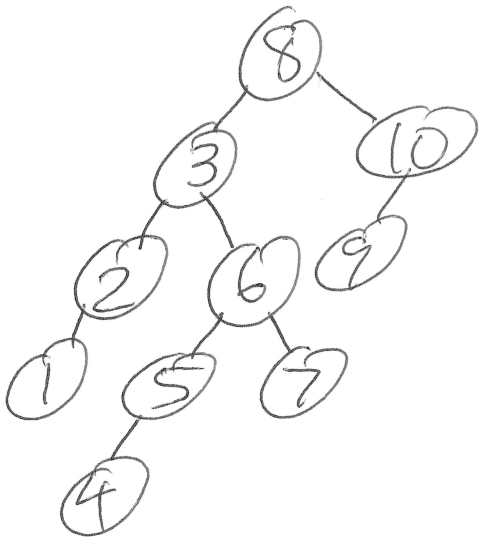
higher (b) least item greater than it

Map

Value BST ⟹ Depth BST

Key Value

8, 3, 6, 2, 5, 7, 1, 4, 10, 9



8 → 0
3 → 1
6 → 2
2 → 2
5 → 3
7 → 3
1 → 3
4 → 4
10 → 1
9 → 2

---

summertrip

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

a b a c d f e g c d f e m n o f

↑ ↑ ⟶

start at c ⟹ find nearest a, b
d, e, f
...

how many ending pts

a → 0, 2,
b → 1
c → 3, 8
d → 4, 9
e → 6, 11
f → 5, 10, 15

TreeSet<Integer>[ ]
idx =

new TreeSet[26];
for (int i=0; i<26; i++)
idx[i] = new TreeSet<Integer>();

```
for (int i=0; i<s.length(); i++)
    idx[s.charAt(i) - 'a'].add(i)
```