COP 3330 11/2/2026

(1) TreeMap

(2) Catching Exceptions

TreeSet - ordered
queries - lower, higher
first, last
] Internally uses a balanced binary search tree

HashMap - each key is mapped to an associated value

	Vote count
Orlando - 0	Bob - 3
Jacksonville - 1	Amy - 7
Tampa - 2	Candace - 2
etc.	etc
ID list	

TreeMap: Ordered keys with associated values!

O(lg n) runtime

ceilingKey(K key)
containsKey(K key)
entrySet()
firstKey
floorKey(K key)
higherKey(K key)
keySet()
lastKey()
lowerKey(K key)
put(Key k, Value v)
remove(Key k)
size()
] Same as HashMap

Don't use containsValue()

VOTES Problem

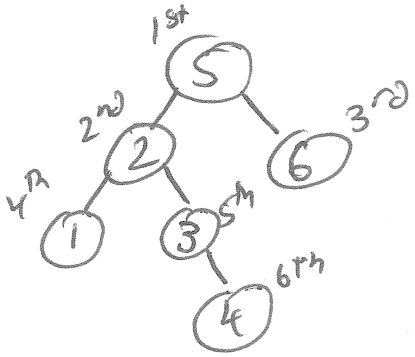
```
1 A
1 B
1 A
1 A
2          ⟶   A 3
                B 1
1 C
1 B
2          ⟶   A 3
                B 2
                C 1
etc.
```

```
TreeMap<String, Integer>
voteMap = new TreeMap<>();
```

```
for (String candidate : voteMap.keySet())
    S.o.p (candidate + ": " + voteMap.get(candidate));
```

Binary Search Tree Problem: Kattis's

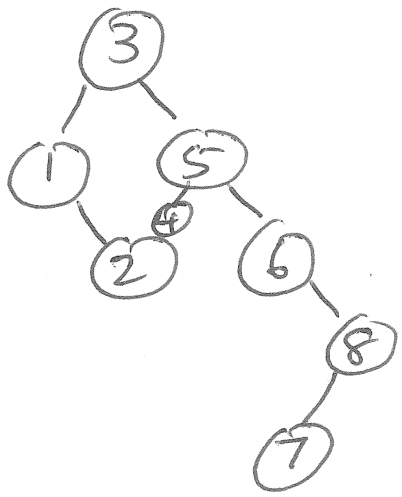
bst insert: 5, 2, 6, 1, 3, 4



Output this case

```
0
1
2
4
6
9
```

Case 2: 3, 5, 1, 6, 8, 7, 2, 4

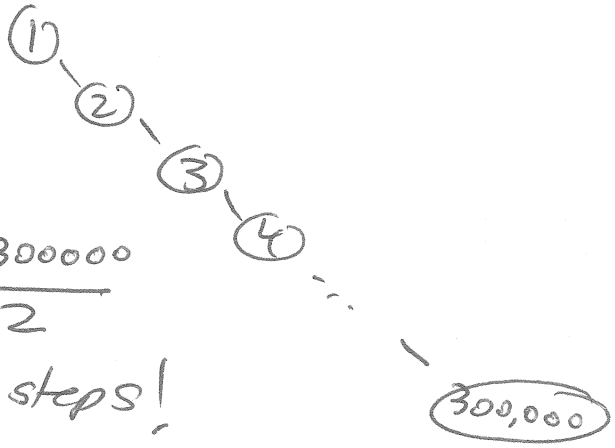


0	+0
1	+1
2	+1
4	+2
7	+3
11	+4
13	+2
15	+2

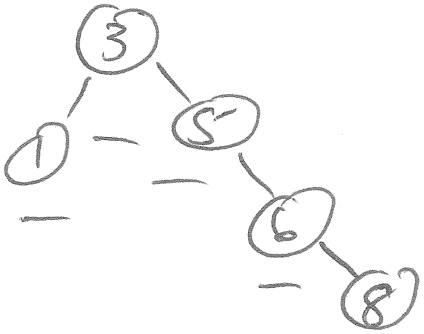
$n \leq 300{,}000$

$1+2+3+4+\ldots+300{,}000 = \frac{299999 \times 300000}{2}$

too many steps!



3, 5, 1, 6, 8, 7



8 has to go right of 6



3 → 0
1 → 1
5 → 1
7 → 4
Value Depth

7 → either right of 6 OR left of 8 → THIS

REASON - greater depth!

map.lowerKey(7) → 6
map.higherKey(7) → 8

One more Priority Queue Example

Sorting Exams

3, 12, 8, 2, 22, 5, 10

15, 8, 2, 22, 5, 10

23, 2, 22, 5, 10

25, 22, 5, 10

47, 5, 10

52, 10

62

15
+23
38
+25
63
+47
110
+52
162
+62
234

2, 3, 5, 8, 10, ~~12~~, 22

5, 5, 8, 10, ~~12~~, 22

8, 10, 10, ~~12~~, 22

10, 12, 18, 22

18, 22, 22

22, 40

5
10
15
+18
33
+18
51
40
91
62
153

Priority Queue -
(1) grab 2 smallest
(2) merge
(3) put back into the queue!

Catching an Exception

new statement: try-catch statement

```
try {
    stmts;
}
catch (InputMismatchException e) {
    S.o.p (e); // prints stack trace
    S.o.p ("try again"
    continue;
}
```

↗ if happens it executes inside this try