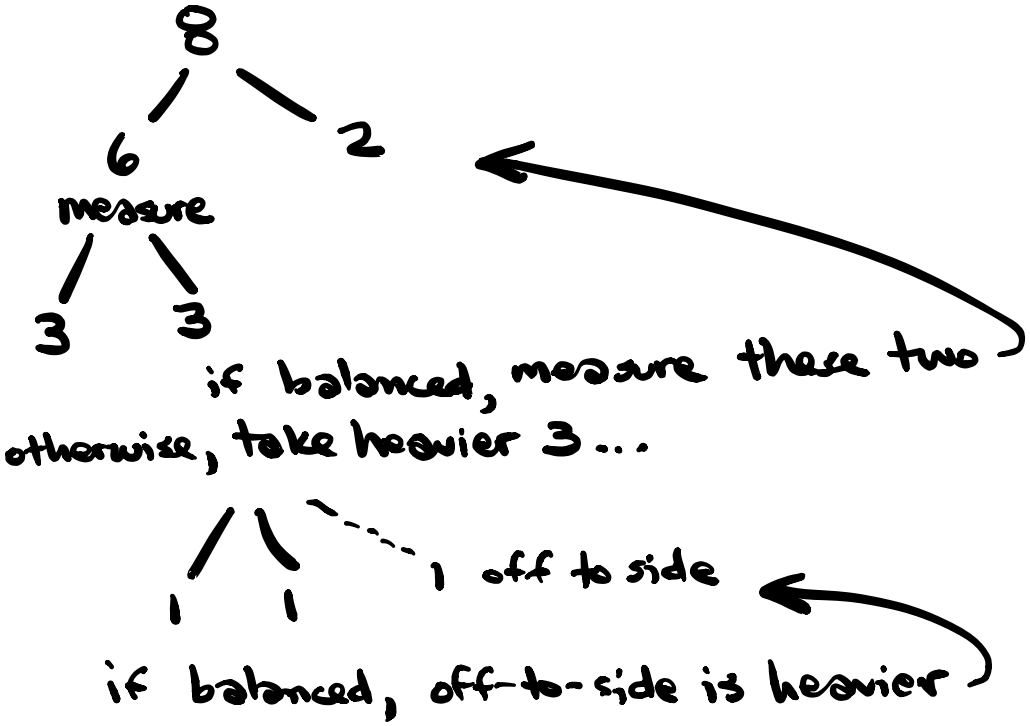
COIN PUZZLE

- 8 coins, 1 is slightly heavier
- how to measure with balance scale to find outlier coin



BIG O NOTATION

$O(n)$ vs $O(n^2)$ vs $O(\log n)$

SEARCH SORTED ARRAY

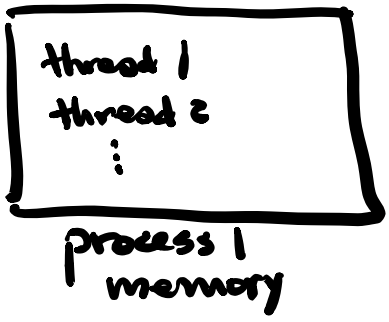
binary search $[O(\log n)]$

L ↑ mid R

is mid == k?
return mid
elif k < mid:

else:
check M+1 → R

THREADS AND PROCESSES



process loads and saves a separate set of registers for each thread

MUTEXES AND SEMAPHORES

mutex → integer 1

lock = decrement
unlock = increment

lock()
unlock()

semaphore → can have values > 1

wait()
signal()

SYNCHRONIZED METHODS IN JAVA

- each object has its own mutex
- synchronized method locks mutex / unlocks at end
- only 1 synchronized method can be called on object at one time

DEADLOCK

thread 1	thread 2
acquire(lock1)	acquire(lock2)
acquire(lock2)	acquire(lock1)

- avoid by requiring locks be acquired in specific order

POLYMORPHISM

- method can have different behavior depending on type of object it is called on or types of parameters

VIRTUAL FUNCTION/METHOD (C++)

A has bar()
↓ subclass
B has bar()

```
A* foo = new B();
foo.bar();
```

- if bar is virtual, this calls B.bar()
  else calls A.bar()

A TO I

Write function to convert string to integer

```
is first char negative sign? if so, save it for later.
total = 0
while ch = next char():
    d = digit(ch)
    total = total * 10 + d
```

beware of overflow if number is too large

REVERSING WORDS IN A STRING

- reverse all chars in string

- find spaces to locate words
- reverse chars in each word

swap chars @ L/R
L++
R--

SORTING

O(n log n)

int[] MergeSort(int[] A)

void QuickSort(int[] A, startIndex, endIndex)

O(n^2) worst-case
O(n log n) average

ORDER STATISTICS

- find kth-largest element
  - select random pivot and partition as in QuickSort
  - based on index of pivot element, we know on which side of pivot element lies

ex: k=15 n=30

after pivot and partition, first half has 10 elements, so 15th is 4th largest in second half; so partition the second-half of the array and continue recursively.

running time is $n + n/2 + n/4 + n/8 + \ldots$

$= O(n)$

- finding median is special case where $k = n/2$

NEAREST NEIGHBOR

each person has name (string) and number (int); the number represents the person's position on a number line.

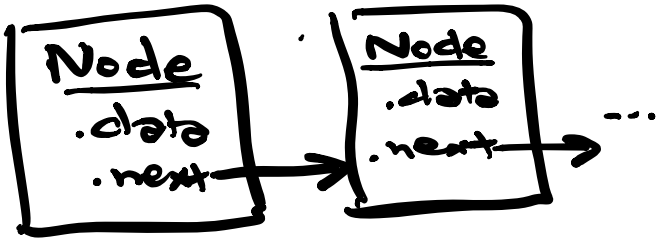
Each person has 3 friends which are the 3 people whose number is nearest their own. Algo to find each person's 3 friends.

A: Sort in ascending order. Check 3 people before and after each person. 3 nearest will be in this group.

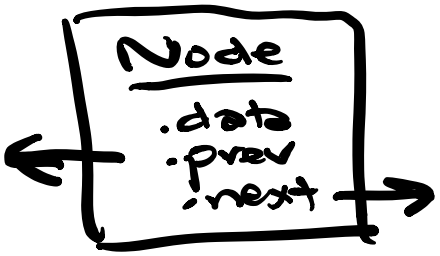
$O(n \log n)$

LINKED LISTS

LinkedList



Doubly Linked List



- adding or removing node is O(1) if we have pointer to the node
- O(N) to look up element and get node ptr

CYCLE IN A LINKED LIST

How do you determine if there is a cycle in a linked list?

- keep track of 2 pointers and start them both at the beginning of the list
- at each iteration move ptr1 by one node and ptr2 by two nodes
- if you reach the end of the list before the ptrs become the same, then there are no cycles

O(n)

STACKS AND QUEUES

Queue — FIFO

Stack — LIFO

HASH TABLES

key → value

(each key has zero or one values)

key → hash function → integer → hash bucket

if collision, then linked list of key/value pairs

insert, remove, lookup each take O(1)

(assuming sufficiently random hash — O(n) worst)

DATA STRUCTURE FOR ANAGRAMS

- for each word, sort letters and use sorted as key in hash table (add word to list which is value of hash table)

sorting → O(n log n)

lookup → O(1)

* if interviewer doesn't like hash tables, you can use tree

FACTORIAL ZEROS

How many zeros are at the end of 100!

100 factorial = 100 · 99 · 98 · 97 · 96 · ...

100 — 1

90 — 1

80 — 1

⋮

10 — 1

= 10

AND

every 2/5 pair

(more 2's than 5's, so just # of 5's)

95 — 1

85 — 1

⋮

5 — 1

= 10

$10 + 10 = 20$

BUT... we forgot the extra 5 for each of 25, 50, 75, 100

$20 + 4 =$ 24 zeroes

(really, just factoring and looking for 2.5 pairs)

DECK SHUFFLING