Complexity ? Efficiency

Motivation

Large Scale
- Healthcare.gov
- Amazon S3
- Fortnite

Smaller Scale
- Slows down analysis

Example: Search (Count # of comparisons)

1	3	5	6	7	8	11	18	14	17	19	21

Sequential Search
Best: 1 test
Worst: n test

Binary Search
Best: 1 test
Worst: 1 data item: 1 test
2 data items: 2 test
4 data items: 3 tests
8 data items: 4 tests
2^k data items: k+1 tests

Trend? # of tests

n tests for 2^{n-1}

Given n items what is the power I take 2 to?

How do I write down $2^? = n$

$? = \log_2 n$

With n items; # of tests is $\log_2 n + 1$

If n is not a power of 2? $\text{floor}(\log_2 n) + 1$

Show google drive.

How do we communicate this? Big-Oh
write big oh for Binary / Sequential search

Big-Oh

Informal Idea: Remove the constants, leave largest growth term

(Example with n^2, n)

$O(1)$	Constant	
$O(\log n)$	Logarithmic	
$O(n)$	linear	
$O(n \log n)$		
$O(n^2)$	quadratic	
$O(n^3)$	cubic	Ex: Matrix Multiplication
$O(2^n)$	exponential	Ex: Fibonacci (actually the constant is smaller), generate subsets
$O(n!)$	factorial	Ex: TSP

Formal

$5n^2 + \frac{n}{3}$ is $O(n^2)$

Means for some value $c > 0$, $c \cdot n^2 \geq 5n^2 + \frac{n}{3}$ once n is big enough

Formal Definition: $f(n)$ is $O(g(n))$ means for some $c > 0$

$c \cdot g(n) \geq f(n)$ for $n \geq k$

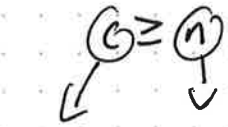
$c \cdot n^2 \geq 5n^2 + \frac{n}{3}$

$c \geq 5 + \frac{1}{3n}$

Choose $c = 6$, $k = 1$

Show $f(n)$ is not $O(g(n))$?

#1 $c \cdot n \geq n^2$ n^2 is to $O(n)$



constant goes to infinity

#2 $c \cdot \log(n) \geq n$ ~~$\log n$~~ n is not $O(\log(n))$

$c \geq \frac{n}{\log(n)}$ ↳ $\lim_{n \to \infty} \frac{n}{\log n} = \lim_{n \to \infty} \frac{1}{\frac{1}{n}} = n$

Stacks

Idea: How do I tell if parentheses are balanced?

$2+(7\cdot 9)/(4+(6+2))$

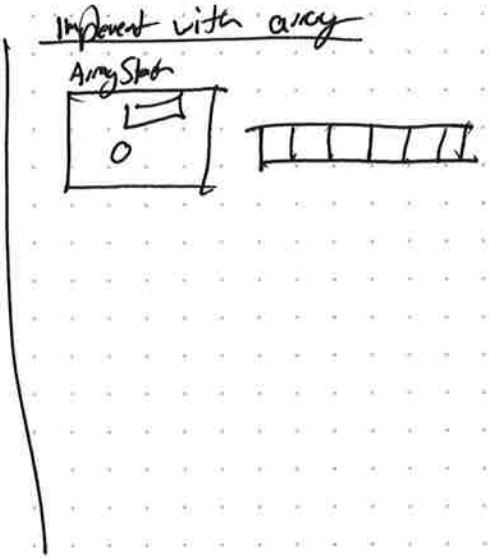
Data Structure: Stack

Stacks

Characteristics: - allows duplicates
- Last in, First out (LIFO) order

Operations: push(T item)
pop()
boolean isEmpty

Implement with array



Ex: Braces and Parentheses

$\{2+(7\cdot 9)/(4+[6+2])\}+9$

push { push (pop) pop } pop ∅

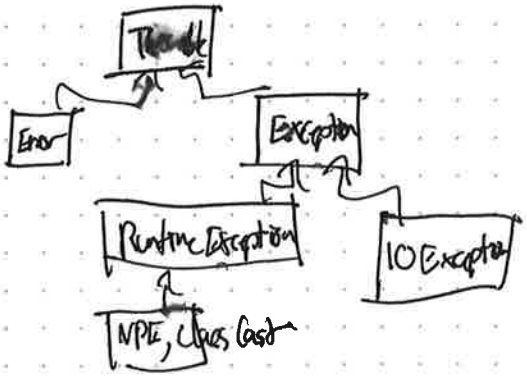
Do array stack from ArrayBag

Exceptions

use throw notation
create our own exception

Checked Exception

Try/catch



Queues

Characteristics:
- allows duplicates
- First in, First out order FIFO

Applications?

Linked Queue

void enqueue add at rear
T dequeue
T getFront
isEmpty

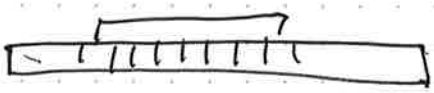
Trick: use Front and Rear pointers

Array Queue

#1 Shift

#2 Track Front

#3 Wrap around



Versions of wrapping

```
if ((rear+1) == max) {
    rear = 0
} else {
    rear++
}
```

→ rear = (rear+1) % max

How do we tell if circular queue is full?

Leave an empty spot ✓

Store a placeholder

Keep a counter

Why do we have to define our CarStack / CarQueue?

Stack ~~/ Queue~~ if old and not written as interfaces

Queue uses... weird... terminology

add/offer, remove/poll