MAKE SCHOOL

Recursion and Search

Check In

MAKE SCHOOL

Share one thing that inspired you this week 

What We're Going to Learn

MAKE SCHOOL

- Understand fundamental search algorithms
- Understand iterative vs. recursive
- Be able to explain recursion at a high level
- Be able to apply recursion to search algorithms

MAKE SCHOOL

Activity: Searching With Cards

Card Searching

MAKE SCHOOL

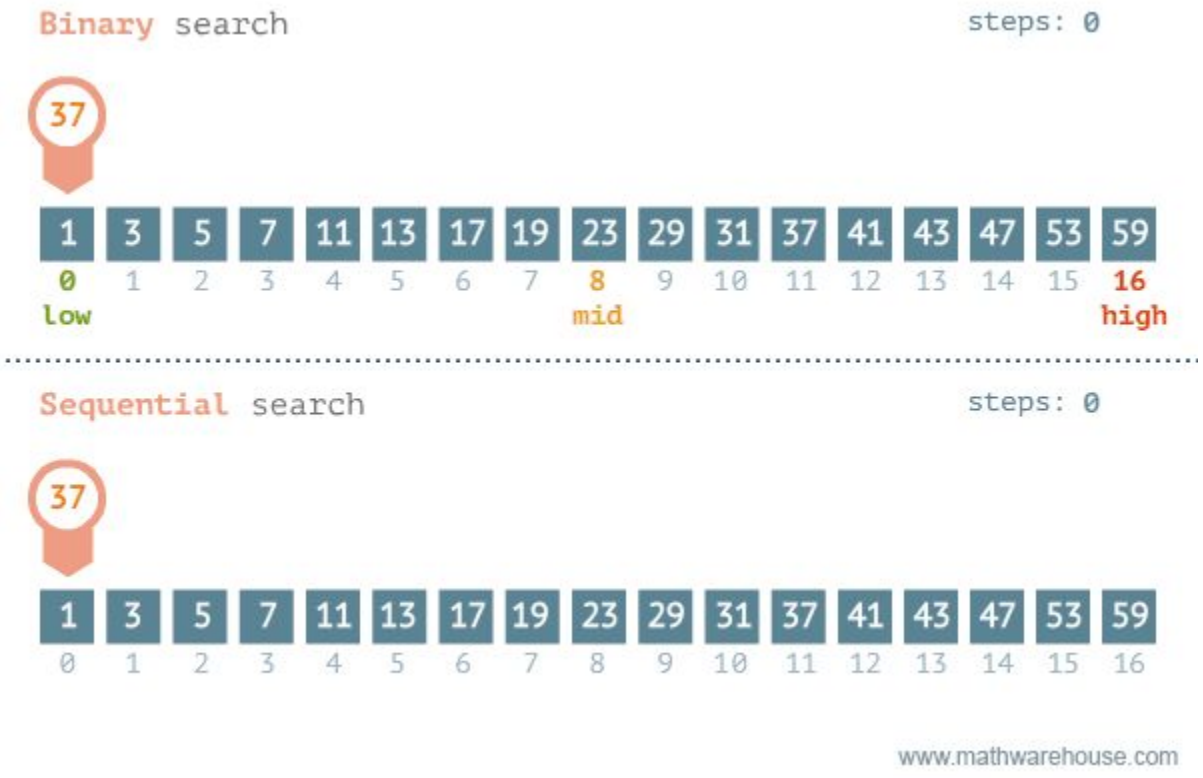
Pull out a deck of cards if you have one otherwise you can use this online tool to help you visualize: https://deck.of.cards/

Given 5 random cards, can you come up with algorithm to search for a target card?

Rule: You can only look at each card one at a time

Students browse: deck.of.cards/

Algorithms Animated



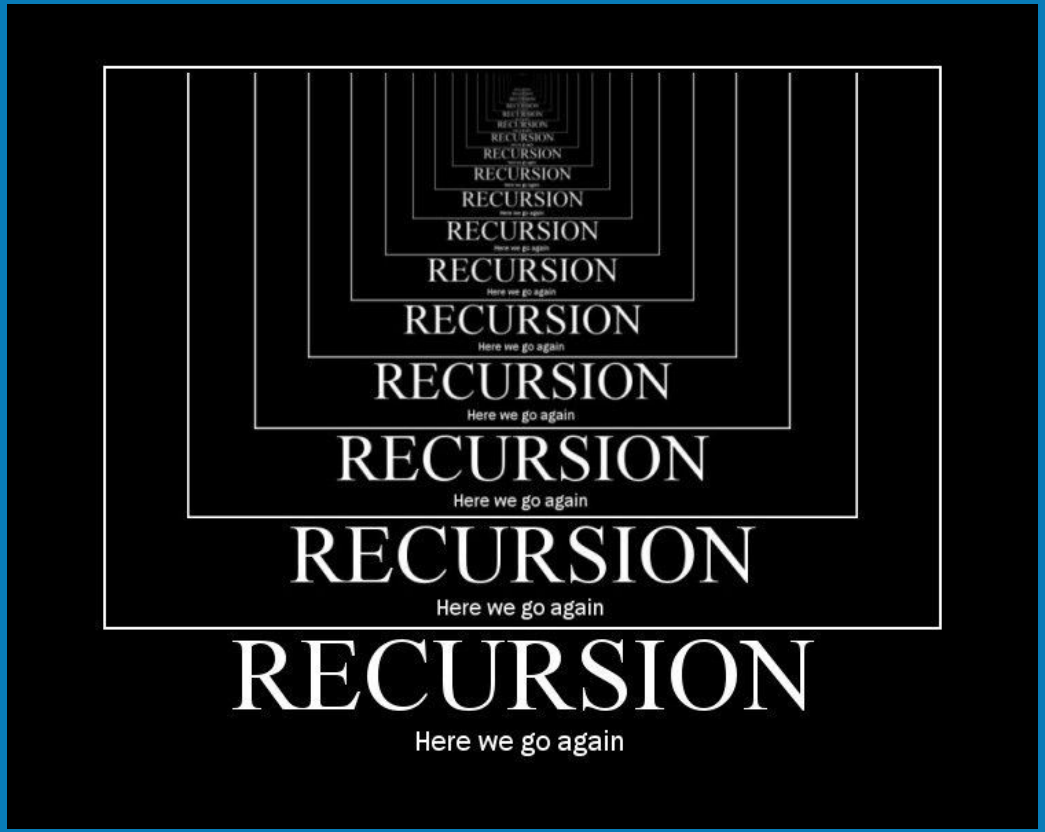
Check Your Understanding

1. In your own words describe the algorithm for sequential search
2. In your own words describe the algorithm for binary search

Students, write your response!

MAKE SCHOOL

Recursion



Factorial

A factorial is when a given number is multiplied by each number less than it.

Use the "!" symbol to represent a factorial

i.e. 4! = 4 * 3 * 2 * 1 = 24

What is 5!

How could we write this more generally?

Factorial

MAKE SCHOOL

n! = n * (n-1) *)n-2) * ... * 2 * 1

n! = n * (n-1)!

You can define factorials in terms of each other!

```
def factorial(n):
    if n == 1:
        return 1
    else:
        f = factorial(n-1)
        return n*f
```

Base Case

Factorial

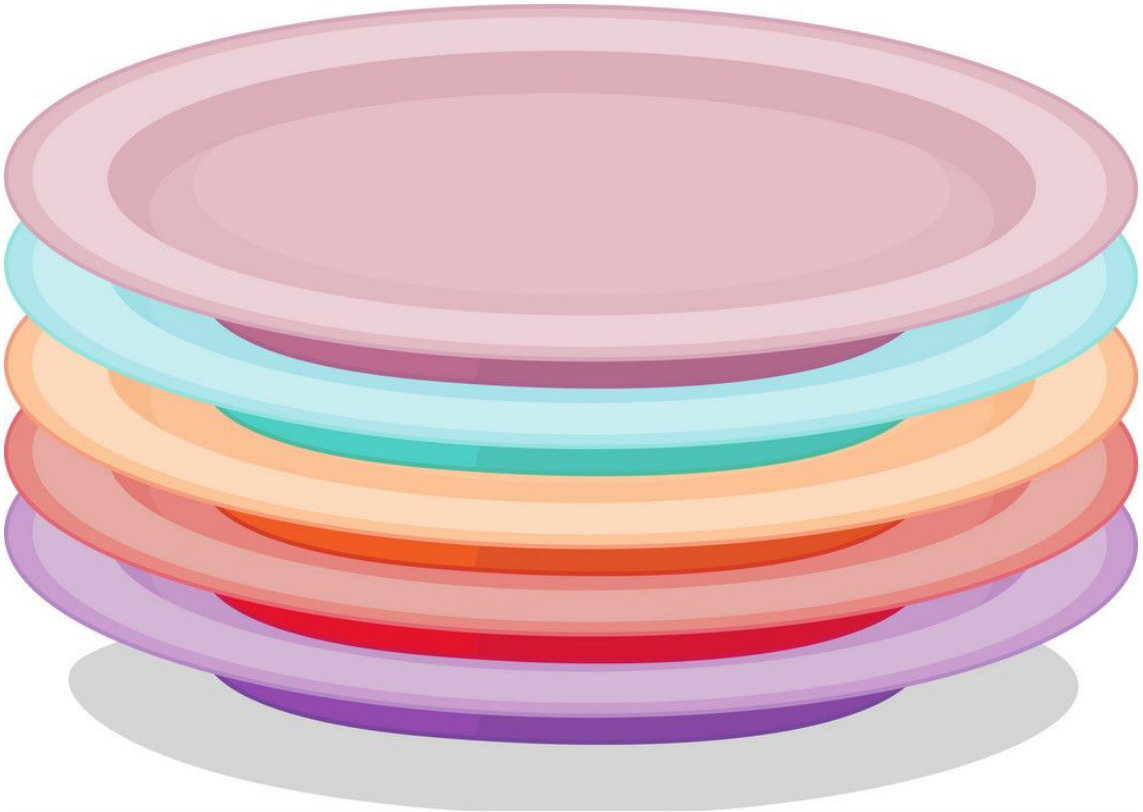
MAKE SCHOOL

Very good resource:

https://www.freecodecamp.org/news/how-recursion-works-explained-with-flowcharts-and-a-video-de61f40cb7f9/

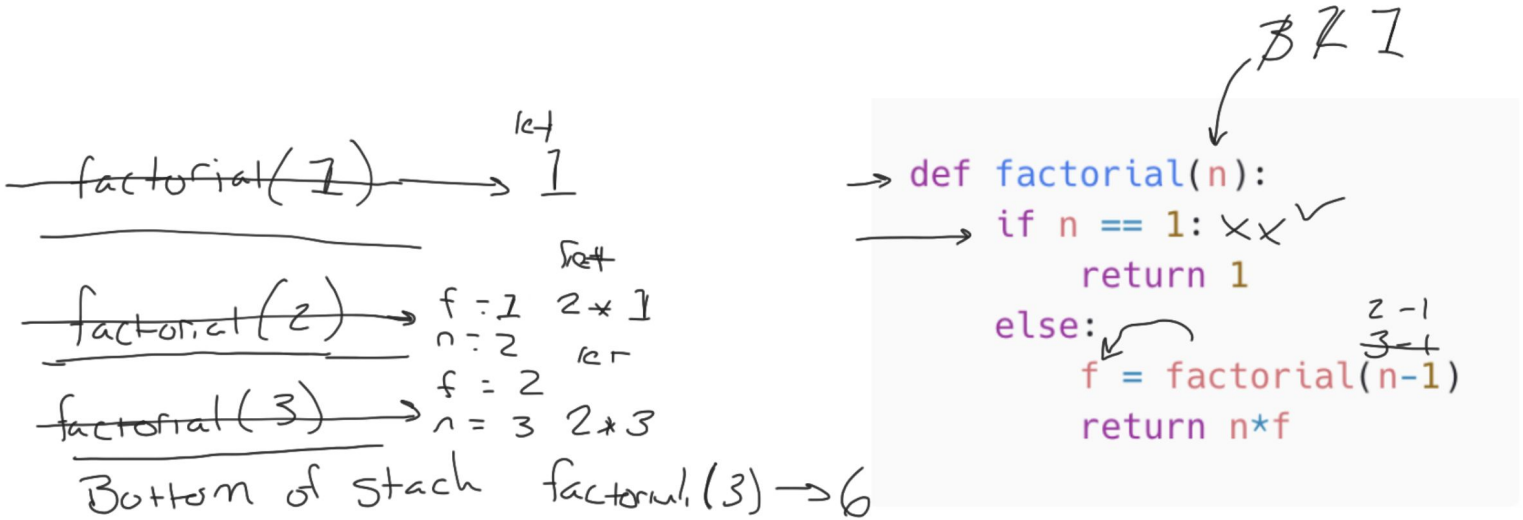
A more concrete way: the call stack

MAKE SCHOOL



Drawing a call stack

Code Trace Call Stack for factorial(3)



Draw the call stack for factorial(2)

MAKE SCHOOL

```
def factorial(n):
    if n == 1:
        return 1
    else:
        f = factorial(n-1)
        return n*f
```

Students, draw anywhere on this slide!

Recursion vs. Iteration

MAKE SCHOOL

- You can rewrite every recursive algorithm as an iterative one
- Recursion isn't always better, it often will use more memory than iteration
- In some cases, recursion may not be as efficient as iteration but it is much more elegant and readable

Iterative Factorial

In your groups think up an algorithm to compute factorial using a loop instead of recursion

Students, write your response!

Let’s turn our algorithm into code!

MAKE SCHOOL

In your groups have one person be the driver and the others be navigators, write the iterative version of factorial

Students browse: repl.it/@MakeSchool/iterativefactorial?lite=true

Recursive Linear Search

MAKE SCHOOL

Get into groups and come up with an algorithm for recursive linear search instead of using a loop

Students, write your response!

Let’s turn our algorithm into code!

In your groups have one person be the driver and the others be navigators, write the recursive version of linear search

Students browse: repl.it/@MakeSchool/recursivelinearsearch?lite=true

MAKE SCHOOL

Iterative Binary Search

Students browse: repl.it/@JessDahmen/iterativebinarysearch?lite=true

MAKE SCHOOL

Recursive Binary Search

Students browse: repl.it/@MakeSchool/recursivebinarysearch?lite=true

Why is Binary Search Log N ?

MAKE SCHOOL

Good article about concept

Check out this article if you want more math

MAKE SCHOOL

Shout Outs

Students, write your response!