Section 2

CSCI E-22

Will Begin Shortly

Practice with Recursion

The following Java method should use recursion to remove all capital letters from a string. Some of the code has been omitted and we will need to fill it in:

```
public static String removeCapitals(String s) {
    if (s == null)
        throw new IllegalArgumentException();

    if (s.equals(""))
        return "";

    String removedFromRest = ____________________;

    ...



}
```

Recall: Recursive Problem Solving

When solving problems using recursion, we break the problem down into smaller subproblems.

Once we have broken down the problem into the smallest subproblem (one that we can solve), we have reached a base case.

Then, we can progressively build up the solutions to the subproblems until we have a solution for the overall problem.

```
sum(n) = n + sum(n-1)
```

Recursive Problem Solving Approach

Before writing a recursive method, we can try to plan:

- What's the base case?
- What's the recursive subproblem?
- What work do we need to do before returning?

The Fibonacci Sequence

The Fibonacci sequence is a well-known number series in which each number in the series is the sum of the two previous numbers.

We define the first two numbers as $F_0 = 0$ and $F_1 = 1$, and all successive numbers as:

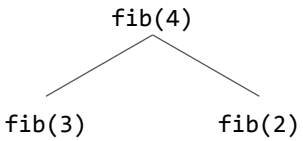
$$F_n = F_{n-1} + F_{n-2}$$

Since the sequence is defined recursively, using recursion to calculate Fibonacci numbers is natural. Finish the code below to write an algorithm that calculates the *n*th Fibonacci number.

```
public static long fib(int n) {

    // base case
    if (_________________)



 }
```

The Fibonacci Sequence

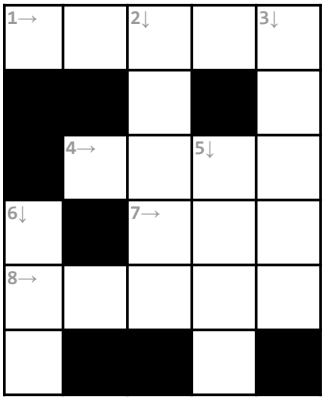
Draw a diagram that shows the number of times `fib()` is called with an initial value of 4. It has been started for you:



Recursive Backtracking

We consider another problem that can be solved with recursive backtracking. Given a list of words that are to be filled into a crossword puzzle, how do we find a solution that satisfies the rules of the conventional crossword?

aft
ale
eel
heel
hike
hoses
keel
laser
lee
line
sails
sheet
steer
tie




```
boolean findSolution(n, other params) {
    if (we found a solution) {
        displaySolution();
        return true;
    }

    for (val = first to last) {
        if (isValid(val)) {
            applyValue(val);
            if (findSolution(n + 1, other params)) {
                return true;
            }
            removeValue(val);
        }
    }

    return false;
}
```