

LEC 10

CSE 123

Exhaustive Search

Questions during Class?
Raise hand or send here

sli.do #cse123



BEFORE WE START

Talk to your neighbors:

*What's your favorite
rainy day activity?*

Instructor: James Wilcox

Announcements

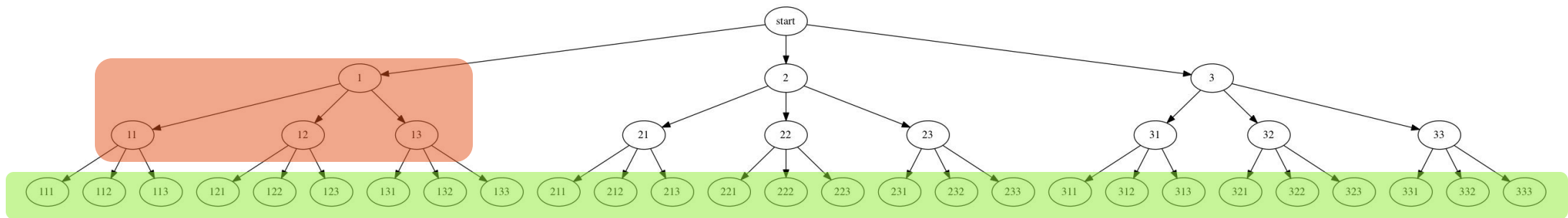
- Good job on Quiz 1!
- Programming Assignment 1 due tonight (10/30) at 11:59pm
- Creative Project 2 released tomorrow, due in one week (11/6)
- Heads up: we did rearrange the last two projects a bit.

Exhaustive Search

- There are some problems computers are bad at solving
 - Polynomial vs. Nondeterministic Polynomial (P vs. NP)
- Password cracking / decrypting is a great example
 - If breaking these were easy, the internet wouldn't be useable
- So what do we do?
 - The stupid way of solving the problem
 - We “exhaustively search” through every possibility
- What do we need? Recursion + String accumulator (public / private pair)

Decision Trees

- Visual we use to help understand what our process is
 - Visualization tool, not a data structure
 - If you can draw a decision tree, you can implement exhaustive search



- Can glean important information
 - **Base case (end nodes)**
 - **Recursive case (middle nodes)**
 - “Dead end” case (more on this later...)

Exhaustive Search Pattern

```
public static void search(input) {
    search(input, "");
}

private static void search(input, String soFar) {
    if (base case) {
        // Do something with soFar (e.g. print it out)
        System.out.println(soFar);
    } else {
        // Might not be a loop, but 1 recursive call for each option
        for (each option) {
            search(input, soFar + option);
        }
    }
}
```

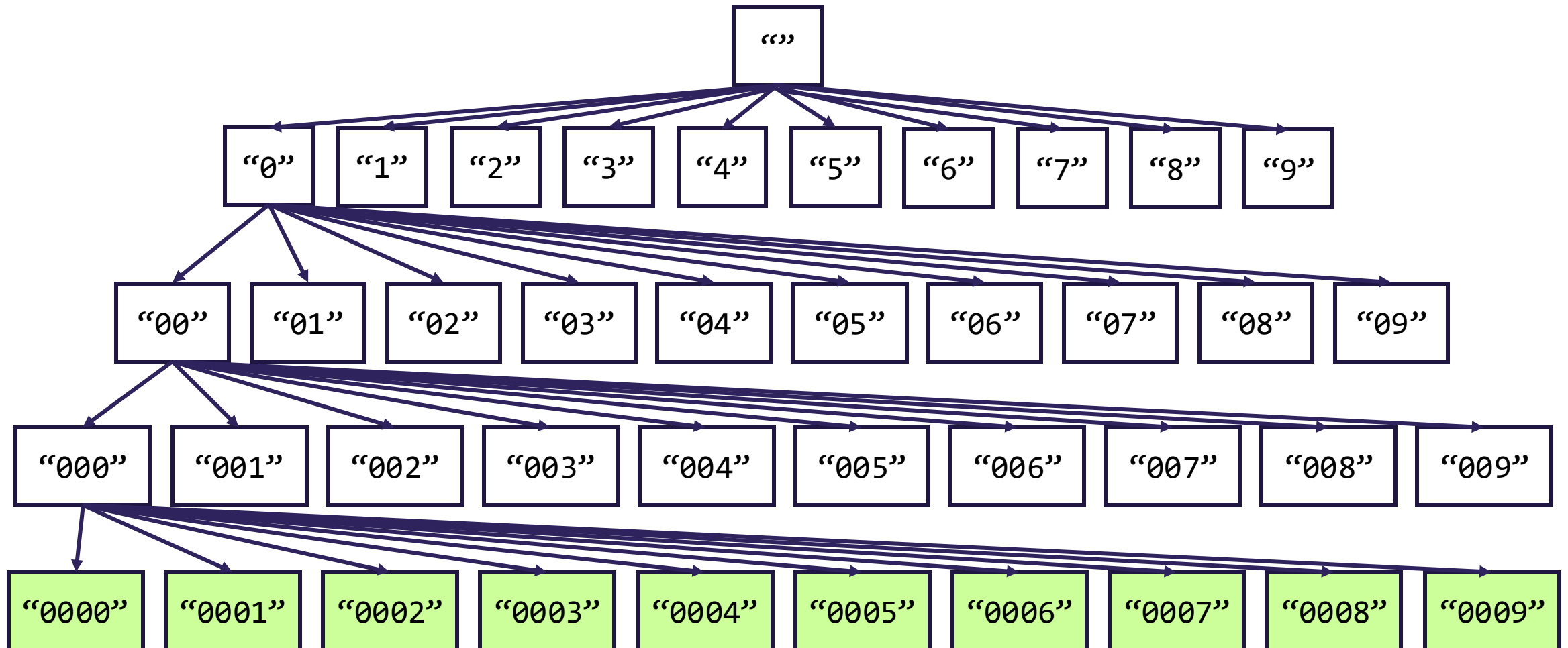
Exhaustive Search Pattern

```
public static void printNums() {
    printNums("");
}

private static void printNums(String soFar) {
    if (soFar.length() == 3) {
        // Do something with soFar (e.g. print it out)
        System.out.println(soFar);
    } else {
        // Might not be a loop, but 1 recursive call for each option
        for (int i = 1; i <= 3; i++) {
            printNums(soFar + i);
        }
    }
}
```

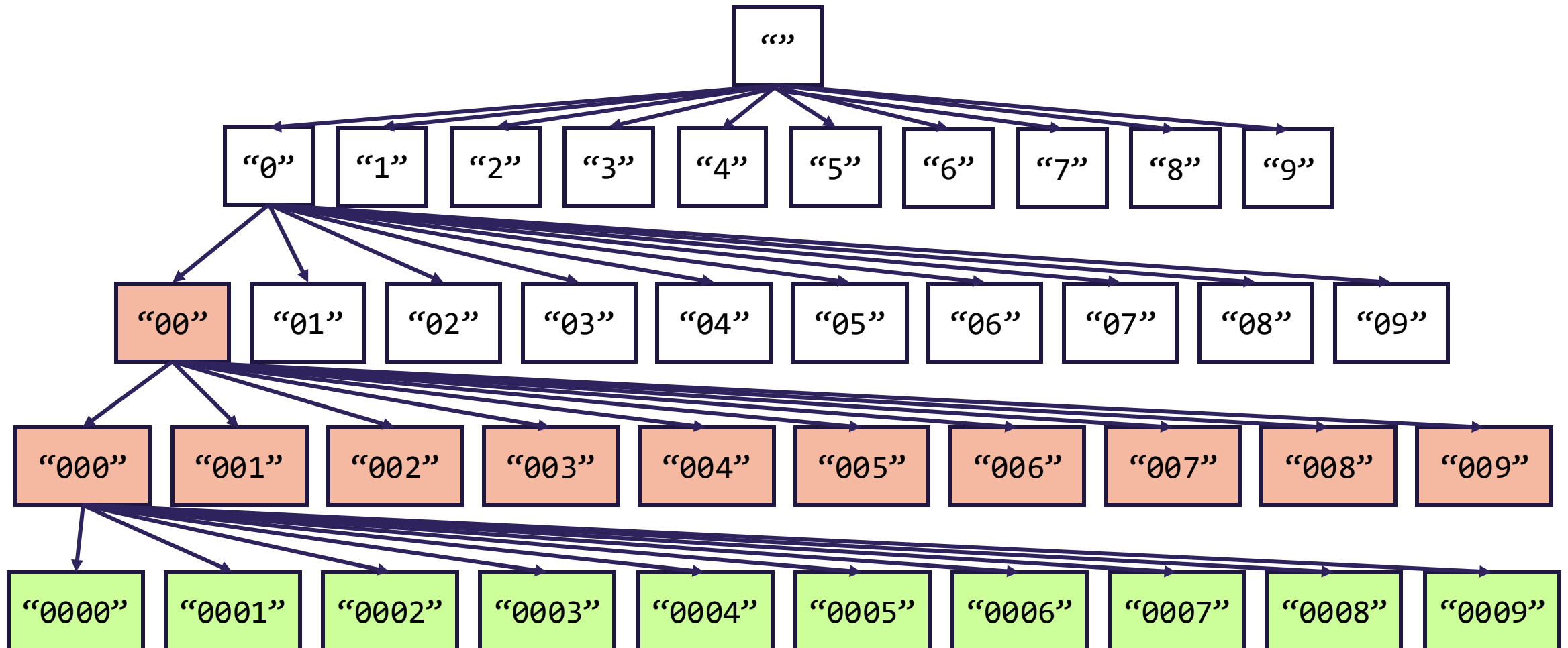

Password Cracker

- Let's say we want to crack the password of a 4 digit combination lock



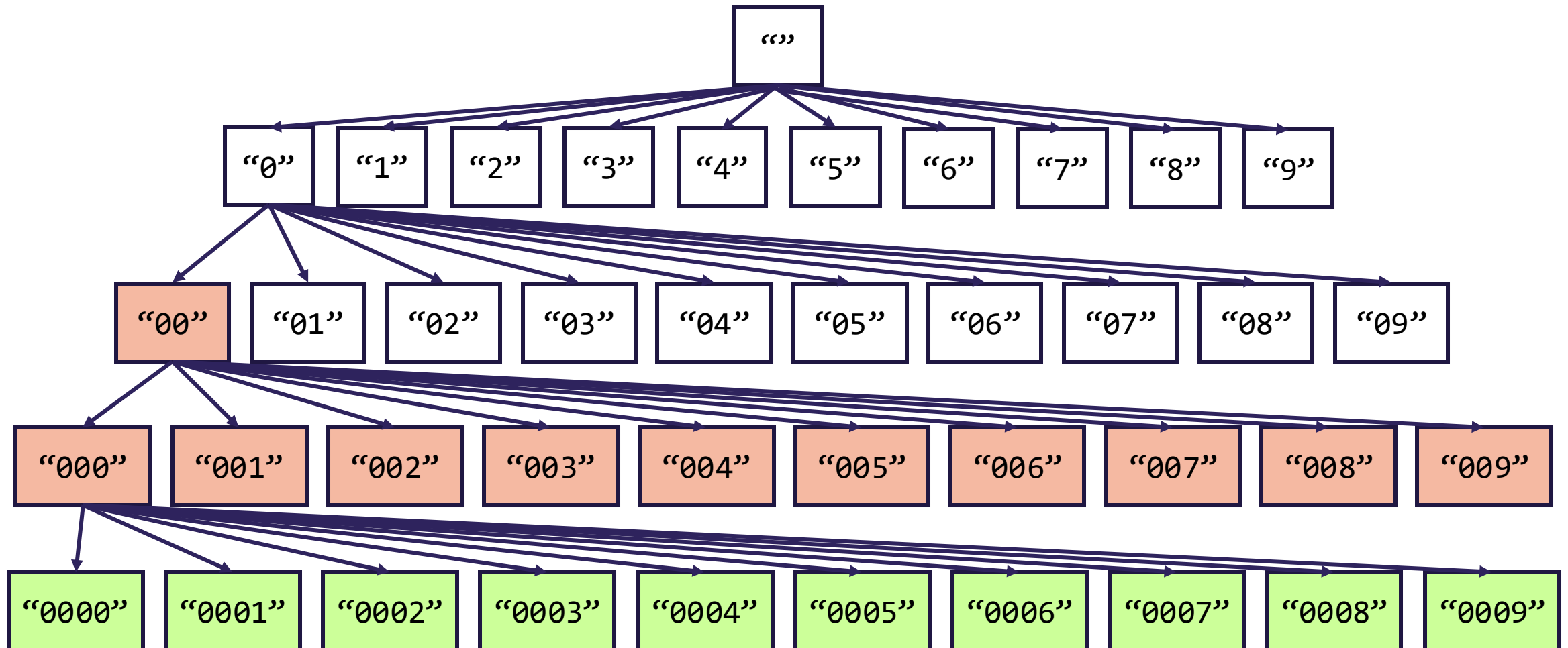
Password Cracker

- Let's say we want to crack the password of a 4 digit combination lock



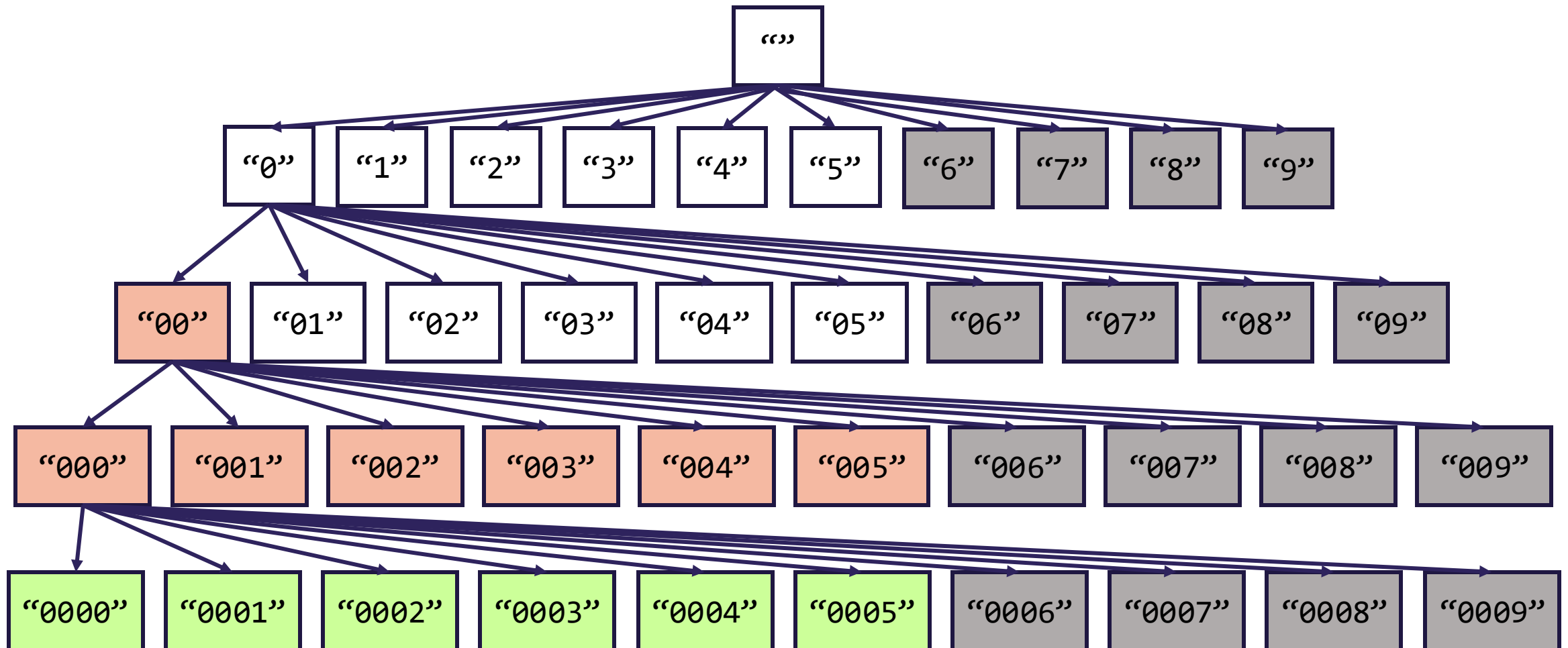
Password Cracker

- Now, what if we knew the sum of all digits was 5?



Password Cracker

- Now, what if we knew the sum of all digits was 5?



Updated Exhaustive Search Pattern

```
public static void search(input) {
    search(input, "");
}

private static void search(input, String soFar) {
    if (base case) {
        // Do something with soFar (e.g. print it out)
        System.out.println(soFar);
    } else if (not dead end) {
        // Might not be a loop, but 1 recursive call for each option
        for (each option) {
            search(input, soFar + option);
        }
    }
}
```