Recap:

Proof by induction

Hypothesis \( P(n) \) for \( n \in \mathbb{Z}^+, \mathbb{N} \)

Inductive step \( P(n) \to P(n+1) \)

Basis step \( P(n_0) \)

\[ \therefore \forall x \, P(x) \text{ for } x \geq n_0 \]

Inductive step \( P(n) \to P(n+1) \)

- Write the l.h.s. of \( P(n+1) \), express it in terms of l.h.s. of \( P(n) \)
- For l.h.s. of \( P(n) \) substitute r.h.s. (assume \( P(n) \))
- Show this results in appropriate r.h.s. expression for \( P(n+1) \)
Recursion: start from biggest element and work down to base case.

"if" statements in alg.

Iteration: start from base case and work up.

"for" statement in alg.

Recall alg. for computing $a^n$

Recursive alg.:

```
rpow(a, n)
if n = 0
  return 1
else
  return a * rpow(a, n-1)
```

Iterative alg.:

```
ipow(a, n)
x = 1
for j = 1, 2, 3, ... n
  x = a * x
return x
```
Binary search (iterative) with a check for $x = a_m$

Binary search $(a, x)$

\[
i \leftarrow 1
\]

\[
j \leftarrow n
\]

\[
m \leftarrow \lfloor (i+j)/2 \rfloor
\]

while $(i < j)$ and $(x \neq a_m)$

\[
\text{if } x > a_m
\]

\[
i \leftarrow m+1
\]

else

\[
j \leftarrow m
\]

endif

\[
m \leftarrow \lfloor (i+j)/2 \rfloor
\]

end while

\[
\text{if } i = m \quad \{ \text{means searched every element}\}
\]

\[
\text{if } (x = a_i)
\]

\[
\text{location} \leftarrow i
\]

else

\[
\text{location} \leftarrow 0
\]

else

\[
\text{location} \leftarrow m
\]

endif

return location
Fibonacci numbers

\[ f_0 = 0 \]
\[ f_1 = 1 \]
\[ f_n = f_{n-1} + f_{n-2}. \]

```
RecurseFibo(n)
    if n == 0
        return 0
    else if n == 1
        return 1
    else
        return RecurseFibo(n-1) + RecurseFibo(n-2)
```

Exponential growth, \( 2^n \), function calls for recursive algorithm
Intuitive example of prod. rule & sum rule

Company makes shirts.

Scenario 1: 2 Ms, 3 Ws.

\# of styles = \# Ms + \# Ws = 5.

Scenario 2: 2 M's, 3 W's, 2 colors for each.

\# of styles = 2 \cdot 2 + 3 \cdot 2 = 10.

Scenario 3: 2 Ms, 3 Ws, 2 colors, 2 size

\# styles: \(2 \cdot 2 \cdot 2 + 3 \cdot 2 \cdot 2 = 20\)
Prod rule

a) # of numbers between 1000 - 9999? (inclusive)

\[ \# = 9 \cdot 10^3 = 9000 \]

b) Do not have repeated digits.

\[ \# = 9 \cdot 9 \cdot 8 \cdot 7 \]

c) How many 1-to-1 func's from domain |m| to |n|

\[ \# : \frac{n!}{m!} \]

e) # of telephone #'s of ten digits  Nxx - Nxx - XXXX

\[ 8 \cdot 10 \cdot 10 \cdot 8 \cdot 10 = 6.4 \times 10^9 \]
**Sum rule** possibilities add.

Two disjoint sets $A_1$ and $A_2$

$$|A_1 \cup A_2| = |A_1| + |A_2| = n_1 + n_2.$$  

Inclusion-exclusion. If $A_1 \cap A_2 \neq \emptyset$

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

Example:

# of bit strings of length 8 that start with 1 or end with 00.

---

# of bit strings starting with 1, \[ B_1 = 2^7 \]

# of bit strings ending with 00, \[ B_{00} = 2^5 \]

# of bit strings start w 1 and ending w 00, \[ B_{100} = 2^5 \]

$$B = B_1 + B_{00} - B_{100}$$

$$= 2^7 + 2^6 - 2^5 = 160.$$
Combining sum & prod.

\[ P_6 = \# \text{ of passwords of length 6, each character an upper-case letter or digit.} \]

\[ P_6 = 36^6 \]

\[ \# \text{ of possible passwords that don't contain a digit.} \]

\[ 26^6 \]

\[ P'_6 = \# \text{ of passwords of length 6 that must contain at least one digit.} \]

\[ 36^6 - 26^6 \]

The pigeonhole principle