Thm: Given an unlimited supply of 5-cent + 7-cent stamps, we can make any amount of postage larger than 23 cents.

**Theorem:** $P(n)$ for every positive integer $n$.

**Proof by induction:** Let $n$ be an arbitrary positive integer. Assume that $P(k)$ is true for every positive integer $k < n$. There are several cases to consider:

- Suppose $n$ is ... *blah blah blah* ...
  Then $P(n)$ is true.

- Suppose $n$ is ... *blah blah blah* ...
  The inductive hypothesis implies that ... *blah blah blah* ...
  Thus, $P(n)$ is true.

In each case, we conclude that $P(n)$ is true.
Proof: Let $n$ be an arbitrary integer $\geq 23$. Assume that for any integer $23 \leq k \leq n$, we can make $k$ cents in postage.

Thus, we can make $n$ cents in postage.
Proof: Let n be an arbitrary int > 23.
Assume that for any int \(23 < k < n\), we can make \(k\) cents in postage.
There are two cases. Either \(n > 28\) or \(n \leq 28\).
Suppose $n > 28$. Then $23 < n - 5 < n$. By IH, we can make $n - 5$ cents in postage. Add a 5-cent stamp for all $n$ cents.

Now suppose $n \leq 28$. ...

In all cases, we can make $n$ cents. □
Proof: Let $n$ be an arbitrary integer $> 23$.

Assume that for any integer $23 < k < n$, we can make $k$ cents in postage.

Six cases to consider:

24 = 7 + 7 + 5 + 5

25 = 5 + 5 + 5 + 5 + 5

26 = 7 + 7 + 7 + 5
27 = 7 + 5 + 5 + 5 + 5
28 = 7 + 7 + 7 + 7
Suppose $n > 28$.
Then $23 < w - 5 < n$.
By IH, we can make $n - 5$ cents in postage.
Add a 5-cent stamp for all $n$ cents.

In all cases, we can make $n$ cents. □
1. Write out the standard template.

**Theorem:** \( P(n) \) for every positive integer \( n \).

**Proof by induction:** Let \( n \) be an arbitrary positive integer. Assume that \( P(k) \) is true for every positive integer \( k < n \).

There are several cases to consider:

- **Suppose** \( n \) is \( \ldots \) \textit{blah blah blah} \( \ldots \)
  
  Then \( P(n) \) is true.

- **Suppose** \( n \) is \( \ldots \) \textit{blah blah blah} \( \ldots \)
  
  The inductive hypothesis implies that \( \ldots \) \textit{blah blah blah} \( \ldots \)
  
  Thus, \( P(n) \) is true.

In each case, we conclude that \( P(n) \) is true.

2. Think big.

But close.
3. Look for holes (base cases).
4. Rewrite everything.
public static int sum(int n) {
  int partialSum;
  partialSum = 0;
  for(int i = 1; i <= n; i++) {
    partialSum += i * i * i;
  }
  return partialSum;
}

Analysis computes \( \sum_{i=1}^{n} i^3 \).

\[
2 + 1 + n + (n+1) + 4 \cdot n = 6n + 4
\]

Tedious!
Which is better?
Running time \( A(n) \leq B(n) \)
when \( n < 20 \)
or \( B(n) < A(n) \)
when \( n \geq 20 \)?

Want to know asymptotic rates of growth for functions
relative rates of growth
$f(n): N \rightarrow \mathbb{R}^+$ be a positive function over real numbers.

Let $g(n): N \rightarrow \mathbb{R}^+$ be another one.

There exist positive constants $c$ and $n_0$ such that $f(n) \leq c g(n)$ for all $n \geq n_0$. 

$O(g(n)) = \{ f(n): \exists \text{ positive constants } c, n_0 \text{ s.t. } f(n) \leq c g(n) \text{ for all } n \geq n_0 \}$
A loose upper bound:

\[ n \leq 256n + 0(n) \]

\[ c = 256, n_0 = 0 \]
$256n = O(n^2)$

$(n_0 = 256, c = 1)$

$n \in O\left(2^{2^2}n\right)$

Let's us compare rates of growth.
\[
\text{"algebra"}
\]
\[
f_1(n) \in O(g_1(n))
\]
\[
f_2(n) \in O(g_2(n))
\]
\[
c \cdot f_1(n) \in O(f_1(n))
\]

for any constant \(c\).

\[
(\Rightarrow \text{for any constant } c, c \in \mathbb{R}(1))
\]
\[ f_1(n) + f_2(n) \in O(g_1(n) + g_2(n)) \]
\[ f_1(n) + f_2(n) \in O(\max\{g_1(n), g_2(n)\}) \]
\[ f_1(n) \cdot f_2(n) \in O(g_1(n) \cdot g_2(n)) \]
```java
class Example {
    public static int sum(int n) {
        int partialSum = 0;
        for(int i = 1; i <= n; i++) {
            partialSum += i * i * i;
        }
        return partialSum;
    }
}
```
for all \( n \geq n_0 \), the same "algebra" rules work! (still a max)

\[
\Omega (g(n(n))) = \Omega (g(n)) = \Omega (g(n)) \\
\Theta (g(n)) = \Theta (g(n)) = \Theta (g(n))
\]
$S(n) \in \Theta(g(n))$ means $g(n)$ is an asymptotically tight bound for $f(n)$
\( o(\log(n)) = \{ f(n) : \exists c > 0, \ \forall n \geq n_0, \ f(n) < c \log(n) \} \)

for all \( n \geq n_0 \),

informally:

\( f(n) \in O(\log(n)) \) but

\( f(n) \notin \Omega(\log(n)) \)
Also there is $w \left( g(n) \right)$ \uparrow \omega

I may say

$f(n) = O(g(n))$

when $f(n) \in O(g(n))$
Function classes $f(n)$ is polynomially bounded if

$$f(n) = O(n^k)$$

for some constant $k$

$$n^{k_1} = o(n^{k_2})$$

when $k_1 < k_2$. 
Exponential functions

\[ n^k = \Theta(a^n) \text{ for any constant } k \neq a > 1. \]

\[ a^n = o(c^n) \text{ for constants } c > a > 1. \]
polylogarithmically bounded functions

\[(\log n)^e = o(n^k)\] for any constants \(e \geq 1\),

\[k > 0.\]

I like to write \(e^{\log_b n} = (\log_b n)^e\).
If $a+b$ are constants

$$\log_a n = \frac{\log_a n}{\log_a b} \in \Theta(\log_a n)$$

$$\Rightarrow \text{Can write } O(\log_a n) \text{ instead of } \Theta(\log_a n)$$

Only if multiplying by $\log!$

$5^{\log_3 n} \in o(5^{1/3})$
Please write things simply as possible.

\( O(n^{\log_3 5}) \) vs. \( O(5^{\log_3 n}) \)

\( O(5n^2) \) vs. \( O(n^2) \)
Asymptotic Notation in Algebra

\( f(n) = \Theta(g(n)) \)

\[ 5n^2 + 1000n = 5n^2 + o(n) \]

\( o(1) \text{ for all choices of functions within each written instance of asymptotic notation on left,} \)
there exists a choice of function for each instance on right, "

\[ f(n) = O(g(Cn)) \]

can choose \( f(n) \) only if \( f(n) \in O(g(Cn)) \)

\[ O(n^2) = O(n^3) \]
0(n^3) \neq O(n^2)

(n^3 \in O(n^3)

n^3 \notin O(n^2))

O(1) + O(n) \cdot O(1) + O(1) = O(n)

\sum n^2 + 1000n = \sum n^2 + O(n)
If somebody writes

\[ \sum n^2 + O(n) \leq \]

must consider all \( f(n) \in O(\alpha) \)

including \( 1000n \).