Prove:

$$1^2 + 2^2 + 3^2 + ... + n^2 = \frac{1}{6}n(n+1)(2n+1)$$

Basically same as:

$$\sum_{i=0}^{n} n^2 = \frac{1}{6}n(n+1)(2n+1)$$

Base Case: n=1

$$0^2 + 1^2 = \frac{1}{6}(1)(1+1)(2(1)+1)$$

$$1 = \frac{1}{6} \cdot 2 \cdot 3 = 1\checkmark$$

Hypothesis:

$\displaystyle\sum_{i=0}^{n} n^2 = \frac{1}{6}n(n+1)(2n+1)$ is TRUE for $n \geq 1$

Proof:

$$\left(\sum_{i=0}^{n} n^2\right) + (n+1)^2 = \frac{1}{6}(n+1)(n+1+1)(2(n+1)+1)$$

Substituting base case in for $\displaystyle\left(\sum_{i=0}^{n} n^2\right)$

$$\frac{1}{6}n(n+1)(2n+1) + (n+1)^2 = \frac{1}{6}(n+1)(n+2)(2n+3)$$

$$\frac{n^3}{3} + \frac{3n^2}{2} + \frac{13n}{6} + 1 = \frac{n^3}{3} + \frac{3n^2}{2} + \frac{13n}{6} + 1\checkmark$$

Polynomial expansion from wolframalpha.com

Prove:
$n < 2^n$ for n > 0

Base Case: n=1
$1 < 2^1 \checkmark$

Hypothesis:
$n < 2^n$ is true for $n \geq 1$

Proof
$(n+1) < 2^{n+1}$

Substitute base case for n

$2^n + 1 < 2(2^n) \checkmark$

Conclusion
Adding one will always be less than multiplying by 2 for 2^n if $n \geq 1$

Prove:

$3^n > 2^n$ for n>0

Base Case: n=1

$3^1 > 2^1 3 > 2$✓

Hypothesis:

$3^n > 2^n$ is true for $n \geq 1$

Proof

$3^{n+1} > 2^{n+1}$

$3^n \cdot 3^1 > 2 \cdot 2^n$

Substitute hypothesis in for 3^n

$3(2^n) > 2(2^n)$✓

Prove:

$9^n - 1$ is a multiple of 8 for all positive integers $n > 0$

Base Case: n=1

$9^1 - 1$

8

$8/8 = 1$✓

Hypothesis

$9^n - 1$ is a multiple of 8 for all positive integers $n \geq 1$

Proof

$9^{n+1} - 1$

$9^n \cdot 9 - 1$

$8(9^n) + 9^n - 1$✓

Conclusion

Anything times 8 is a multiple of 8, so $8(9^n)$ is a multiple of 8. In the base case, we already proved $9^n - 1$ is a multiple of 8

Prove

$4^n - 1$ is a multiple of 3 for all positive integers $n > 0$

Base Case: n=1

$4^1 - 1$

3

$3/3 = 1$✓

Hypothesis

$4^n - 1$ is a multiple of 3 for all positive integers $n \geq 1$

Proof

$4^{n+1} - 1$

$4n \cdot 4^1 - 1$

$3(4n) + 4n - 1$✓

Conclusion:

Since $4n$ is being multiplied by 3, that term has to be a multiple of 3. We already proved that $4n - 1$ is a multiple of 3 in the base case.

Prove:

```
def func(n):
    i = 0
    if (n > 1):
        func(n - 1)
    for i in range(n):
        print(" * ", end=" ")
```

Base Case: n=1

```
func(1):
    i = 0
    if (1 > 1) # False:

    for i in range(1):
        print(" * ", end=" ")
```

Assume

`func(n)` is true for $n \geq 1$

Proof

```
func(n+1):
    i = 0
    if (n+1 > 1): #True
        func(n + 1 - 1) # Goes back to assumption
```

Conclusion

n + 1 - 1 = n, so it just goes back to our assumption

Prove:

```
def func(array, n):
    if (n==1):
        return array[0]
    else:
        x = func(a, n-1)
    if (x > a[n-1]):
        return x
    else:
        return a[n-1]
```

Base Case: n=1

```
func(array, 1):
    if (1 == 1): # True
        return array[0]
```

Assumption:

`func(n)` is true for $n \geq 1$

Proof:

```
func(array, n+1):
    if (n + 1 == 1): # False
    else:
        x = func(a, n + 1 - 1) # Goes back to assumption
```