HANDONG GLOBAL UNIVERSITY

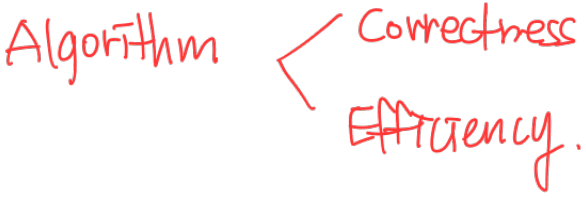
Chapter 4
Divide and Conquer

Algorithm Analysis

School of CSEE

Recurrence Equation

- In many cases we can use a ***recurrence equation*** to describe the running time of an algorithm.
- In this chapter we will study how to solve recurrence equation.



Recurrences

- **The expression:**

$$T(n) = \begin{cases} c & n = 1 \\ 2T\left(\frac{n}{2}\right) + cn & n > 1 \end{cases}$$

is a *recurrence*.

  - **Recurrence: an equation that describes a function in terms of its value on smaller functions.**

- **Technicalities**
  - **Floors and ceilings**
  - **Exact vs. asymptotic functions**
  - **Boundary conditions**

Recurrence Examples

$$s(n) = \begin{cases} 0 & n = 0 \\ c + s(n-1) & n > 0 \end{cases}$$

base case

$$s(n) = \begin{cases} 0 & n = 0 \\ n + s(n-1) & n > 0 \end{cases}$$

$$T(n) = \begin{cases} c & n = 1 \\ 2T\left(\frac{n}{2}\right) + c & n > 1 \end{cases}$$

$$T(n) = \begin{cases} c & n = 1 \\ aT\left(\frac{n}{b}\right) + cn & n > 1 \end{cases}$$

$b > 1$

Solving Recurrence Equations

- Substitution method
- Recursion-tree method
- Iteration method
- Master method

Substitution method

- Guess the form of the solution.
- Prove the guess is correct by mathematical induction.

Substitution method

$$T(n) = \begin{cases} \Theta(1) & \text{if} \quad n = 1 \\ 2T(\frac{n}{2}) + n & \text{if} \quad n > 1 \end{cases}$$

merge sort의 recurrence equation.

Guess $T(n) = O(n \lg n)$

Let us prove that $T(n) \le cn \lg n$ for some c>0.

$$T(n) = 2T(\frac{n}{2}) + n$$

$$\le 2c\frac{n}{2}\lg\frac{n}{2} + n$$

$$= cn\lg n - cn + n$$

$$= cn\lg n - (c-1)n$$

$$\le cn\lg n \quad \text{for} \quad c \ge 1$$

$T(n) = 2T(\frac{n}{2}) + n$

$\le 2c\frac{n}{2}\lg\frac{n}{2} + n$

$= 2c\frac{n}{2}(\lg n - \lg 2) + n$

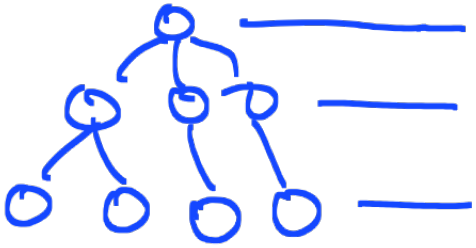
$= cn\lg n - cn\lg 2 + n$

$= cn\lg n - cn + n$

$= cn\lg n - (c-1)n.$

Recursion-tree method

- Build a recursion tree.
  - Each node is the cost of a single subproblem.
- Sum the costs within each level.
- Sum the costs of all levels.

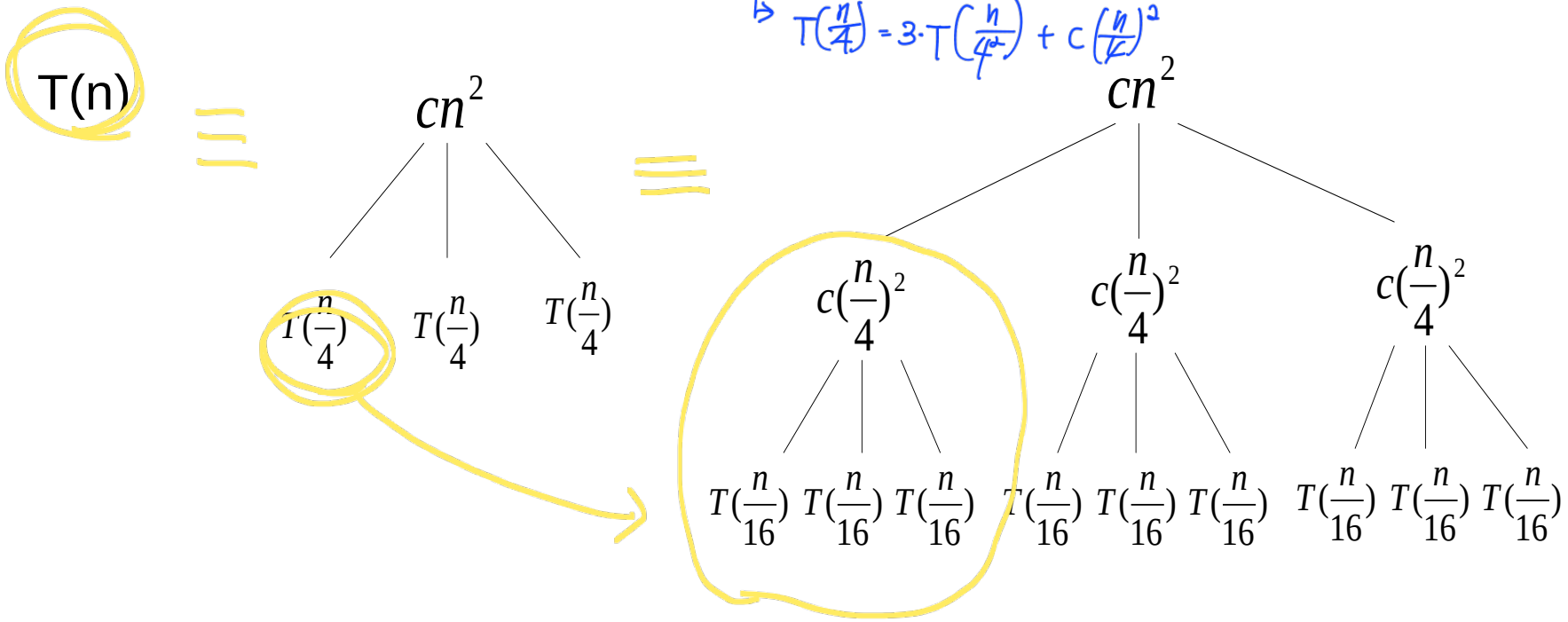


level마다 cost를 더해서
모든 각각의 cost를 더하면
전체의 합을 구할수 있음.

Construction of a recursion tree

$$T(n) = 3T(\frac{n}{4}) + cn^2$$

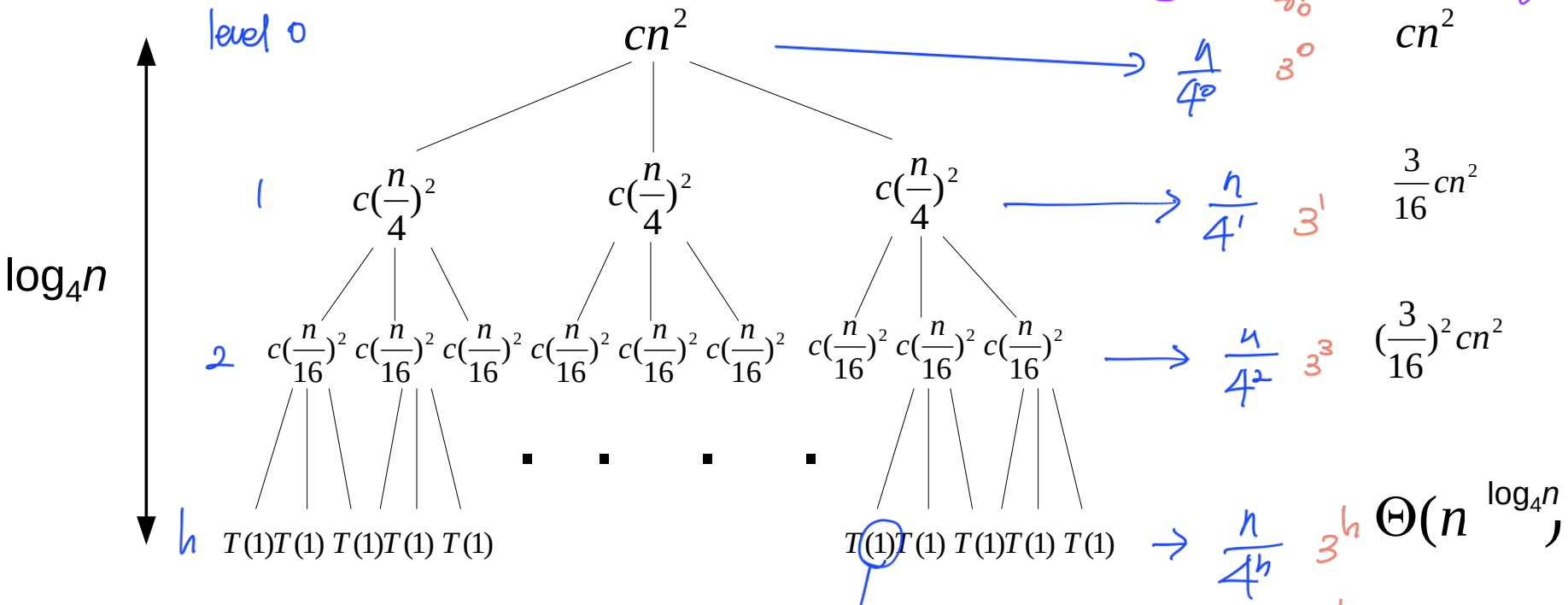
$T(\frac{n}{4}) = 3 \cdot T(\frac{n}{4^2}) + c(\frac{n}{4})^2$



Construction of a recursion tree

$$T(n) = 3T(\frac{n}{4}) + cn^2$$

① height를 구하고 ($h = \log_4 n$)
② 잎의 개수 구하기 ($n^{\log_4 3}$)
③ Total 값 구하기. (next page)

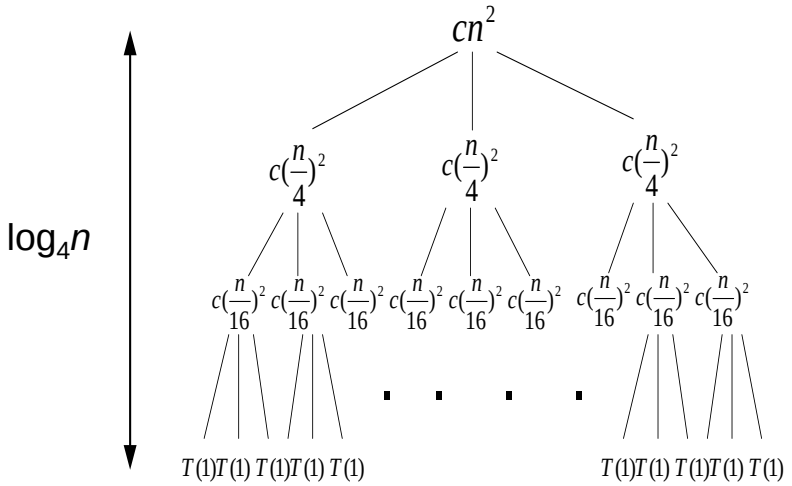


$\frac{n}{4^h} = 1$, $4^h = n$, $h = \log_4 n$

↳ $3^{\log_4 n} = n^{\log_4 3}$

Construction of a recursion tree

$$T(n) = 3T(\frac{n}{4}) + cn^2$$



cn^2

$\frac{3}{16}cn^2$

$(\frac{3}{16})^2 cn^2$ — $\log_4 n + 1$

$\Theta(n^{\ln_4 3})$ — $\log_4 n$

$$= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + ... \left(\frac{3}{16}\right)^{\ln_4 n - 1} cn^2 + \Theta(n^{\ln_4 3})$$

Recursion-tree method

$$T(n) = \underbrace{cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \ldots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2}_{\text{↳ Internal nodes}} + \underbrace{\Theta(n^{\log_4 3})}_{\text{↳ a leaf node}}$$

height − 1 (leaf node)

$$= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})$$

i번째 term

$$= \frac{1 - \left(\frac{3}{16}\right)^{\log_4 n}}{1 - \frac{3}{16}} cn^2 + \Theta(n^{\log_4 3})$$

$\left(\frac{1-r^n}{1-r}\right)a$

$$= \frac{16}{13}cn^2 - \frac{16}{13}\left(\frac{3}{16}\right)^{\log_4 n} cn^2 + \Theta(n^{\log_4 3})$$

$$< \frac{16}{13}cn^2 + \Theta(n^{\log_4 3})$$

$$= O(n^2)$$

$\left(\frac{3}{16}\right)^{\log_4 n} = n^{\log_4 \frac{3}{16}}$ ← # 1보다 작으니까 negative

$= n^{-0.xxx}$ 인데

$n^{-0.xxx} * cn^2$ 하면 제일큰 term보다 작아지잖아.

T(n) = Θ(n²) since T(n) >= cn²

$T(n) = 3T\left(\frac{n}{4}\right) + cn^2$ 이면 cn^2보다 큼.

00:07:14.00

Algorithm Analysis Chapter 4 12

21:01 / 30:56 x2.0

d_1_alg_ch4_1

한동대학교 HANDONG GLOBAL UNIVERSITY School of CSEE

$$T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 \ldots + \left(\frac{3}{16}\right)^{\log_4 n} \cdot cn^2$$

$$= \frac{1 - \left(\frac{3}{16}\right)^{\log_4 n + 1}}{1 - \frac{3}{16}} cn^2 \quad ?$$

$$< cn^2$$

$$= \frac{16}{13}\left(1 - \frac{3}{16}\right)\left(\frac{3}{16}\right)^{\log_4 n} cn^2$$

$$= O(n^2)$$

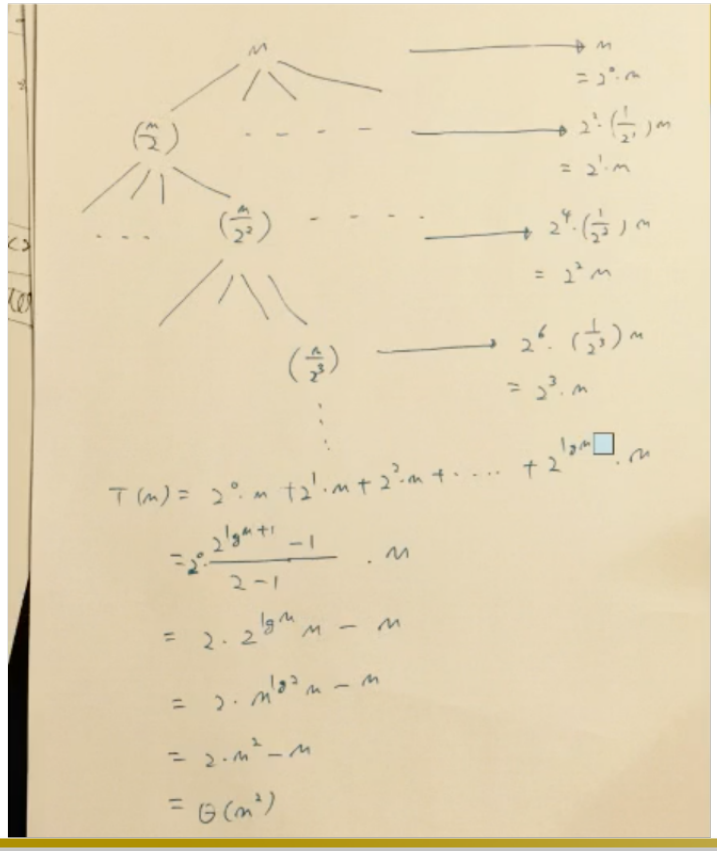
00:13:16.00

Algorithm Analysis Chapter 4 12

28:54 / 30:56 x2.0

Exercise

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$



Iteration Method

- Another option is the “iteration method”
  - Expand the recurrence
  - Work some algebra to express as a summation
  - Evaluate the summation

- We will show a few examples.

Example

- $s(n)$
  $= c + s(n\text{-}1)$
  $= c + c + s(n\text{-}2)$
  $= 2c + s(n\text{-}2)$
  $= 2c + c + s(n\text{-}3)$
  $= 3c + s(n\text{-}3)$
  …
  $= kc + s(n\text{-}k) = ck + s(n\text{-}k)$

$$s(n) = \begin{cases} 0 & n = 0 \\ c + s(n-1) & n > 0 \end{cases}$$

way 2.

$s(n) = c + s(n-1)$
$s(n-1) = c + s(n-2)$
$s(n-2) = c + s(n-3)$
+
$s(1) = c + s(0)$

$s(n) = c \cdot n + s(0) = cn$

$\therefore s(n) = \Theta(n)$

Example

- So far for $n >= k$ we have
  - $s(n) = ck + s(n-k)$
- What if $k = n$?
  - $s(n) = cn + s(0) = cn$
- Thus in general
  - $s(n) = cn = \theta(n)$

$s(n) = (n + s(m-n)) = cn + s(0)$

Exercise

$$s(n) = \begin{cases} 0 & n = 0 \\ n + s(n-1) & n > 0 \end{cases}$$

Way 1

- $s(n)$
  $= n + s(n-1)$
  $= n + n-1 + s(n-2)$
  $= n + n-1 + n-2 + s(n-3)$
  $= n + n-1 + n-2 + n-3 + s(n-4)$
  $= \cdots$
  $= n + n-1 + n-2 + n-3 + \cdots + n-(k-1) + s(n-k)$
  $= \sum_{i=n-k+1}^{n} i + s(n-k)$
- so far for $n \geq k$ we have $\sum_{i=n-k+1}^{n} i + s(n-k)$
- What if $k = n$?

$$\sum_{i=1}^{n} i + s(0) = \sum_{i=1}^{n} i + 0 = n\frac{n+1}{2} \qquad \therefore s(n) = \theta(n^2)$$

Way 2

$s(n) = n + s(n-1)$
$s(n-1) = (n-1) + s(n-2)$
$s(n-2) = (n-2) + s(n-3)$
$\vdots$
$s(1) = 1 + s(0)$

$+$

$s(n) = 1 + 2 + \cdots + n + s(0)$
$= \frac{1}{2}n(n+1) = \theta(n^2)$.

Master method

Master theorem

Let $a \geq 1$, $b > 1$ be constants, function $f(n) > 0$ and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = a\,T(n/b) + f(n)$$

Then $T(n)$ can be bounded asymptotically as follows.

1.If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.

2.If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.

3.If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$, and if $f(n)$ satisfies the regularity condition $a\,f(n/b) \leq c\,f(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Intuition

- Compare $f(n)$ with $n^b \log_a n$

- Not all the possibilities for $f(n)$ are covered in three cases.

Exercise 1

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

① 어떤 case에 속하는지 확인하기

$a=4 \quad b=2$

$n^{\log_b a} = n^2 \qquad f(n)=n.$

$f(n) = O(n^{2-\varepsilon})$ for $\varepsilon=1$

This is the case 1. So we have.

$T(n) = \Theta(n^2)$

$n^{\log_2 4} = n^2$

$n^{\log_2 2^2} = n^{2\log_2 2} = n^2$

$\varepsilon=1$

$O(n^1)$

$\varepsilon=\frac{1}{2}$

$f(n) \le O(n^{\frac{3}{2}})$

Exercise 2

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$a = 4 \qquad b = 2$

$n^{\log_b a} = n^2 \qquad f(n) = n^2$

$f(n) = \Theta(n^2)$

This is the case 2. So we have

$T(n) = \Theta(n^2 \lg n)$

Exercise 3

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

$a=4 \quad b=2$

$n^{\log_b a} = n^2 \qquad f(n) = n^3$

$f(n) = \Omega(n^{2+\varepsilon})$ for $\varepsilon = 1$

만족하는지 확인하기 → $af(n/b) \le cf(n)$

$4\left(\frac{n}{2}\right)^3 \le cn^3$ for $c = \frac{1}{2}$

확인 why 하는가?

This is the case 3. So we have

$T(n) = \Theta(n^3)$

Proof By Induction

- Base case: n=0 or n=1
  - Show formula is true when n = 0 or 1
- Inductive hypothesis:
  - For n greater than 0, assume that the formula holds true for all $k \geq 0$ such that $k < n$
  - By inductive hypothesis, the formula is true when $k = n - 1$
- Proof of goal statement:
  - Show that formula is then true for n

Example: Gaussian Closed Form

- Prove $1 + 2 + 3 + \ldots + n = n(n+1) / 2$
  - Base case: when $n=1$, then $1 = 1(1+1)/2$
  - Inductive hypothesis:
    - For n greater than 0, assume that $1+2+3+\ldots+k = \frac{k(k+1)}{2}$ holds true all $k \geq 0$ such that $k < n$.
    - By hypothesis the formula is true when $k = n-1$, $1+2+3+\cdots+(n-1) = n(n-1)/2$
  - Proof of goal statement:

$$1+2+\cdots+(n-1)+n = 1+2+\cdots+(n-1)+n$$

$$= \frac{n(n-1)}{2} + n = (n)(n+1)/2$$

Exercise

- Prove $1^2 + 2^2 + 3^2 + \ldots + n^2 = n(n+1)(2n+1) / 6$
  - Base case:
    - When $n = 1$, then $1 = 1(1+1)(2+1) / 6$ # left = right 같음을 증명.
  - Inductive hypothesis:
    - For n greater than 0, assume that $1^2 + 2^2 + 3^2 + \ldots + k^2 = k(k+1)(2k+1) / 6$ holds true all $k \geq 0$ such that $k < n$.
    - By hypothesis the formula is true when $k = n-1$,

      $1^2 + 2^2 + 3^2 + \ldots + (n-1)^2 = (n-1)n(2n-1) / 6$

  - Proof of goal statement:

    $1^2 + 2^2 + \ldots + (n-1)^2 + n^2 = (1^2 + 2^2 + \ldots + (n-1)^2) + n^2$

    $= (n-1)n(2n-1) / 6 + n^2 = n(n+1)(2n+1) / 6$