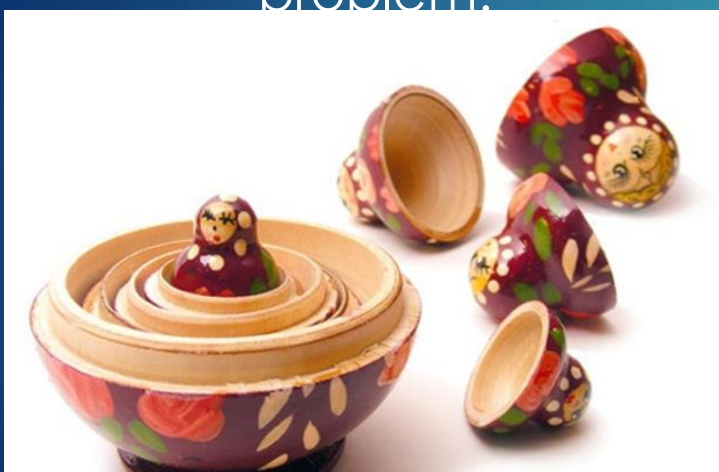
RECURSIVE

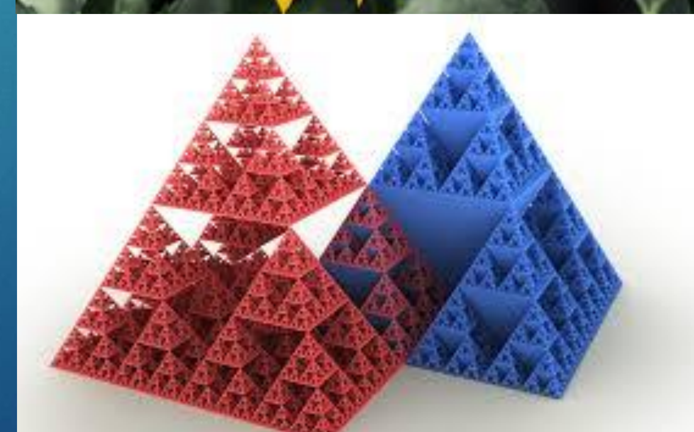
DATA STUCTURE

ThS.Trần Lê Như Quỳnh

DEFINITION

- **T**he solution of a big problem depends on solutions to smaller instances of the same problem.





ThS.Trần Lê Như Qu

HOW TO SOLVE PROBLEM BY RECURSIVE

ThS.Trần Lê Như Quỳnh

- if problem is "small enough"
- solve it directly
- else
- break into one or more smaller sub-problems
- solve each sub-problem recursively
- combine results into solution to whole problem

BASIC EXERCISES

RECURSIVE

ThS.Trần Lê Như Quỳnh

ThS.Trần Lê Như Quỳnh

FACTORIAL

BASIC RECURSIVE EXERCISES

FACTORIAL DEFINITION

ThS.Trần Lê Như Quỳnh

- 0! = 1
- 1! = 1 * 1
- 2! = 1 * 2
- 3! = 1 * 2 * 3
- …
- N! = 1 * 2 * 3… * N

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 & \text{if } n \geq 1. \end{cases}$$

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot (n-1)! & \text{if } n \geq 1. \end{cases}$$

FACTORIAL IMPLEMENT

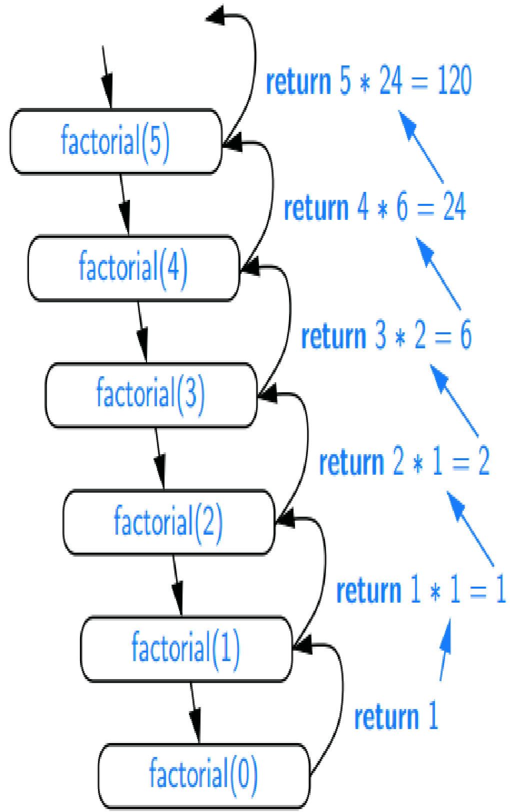


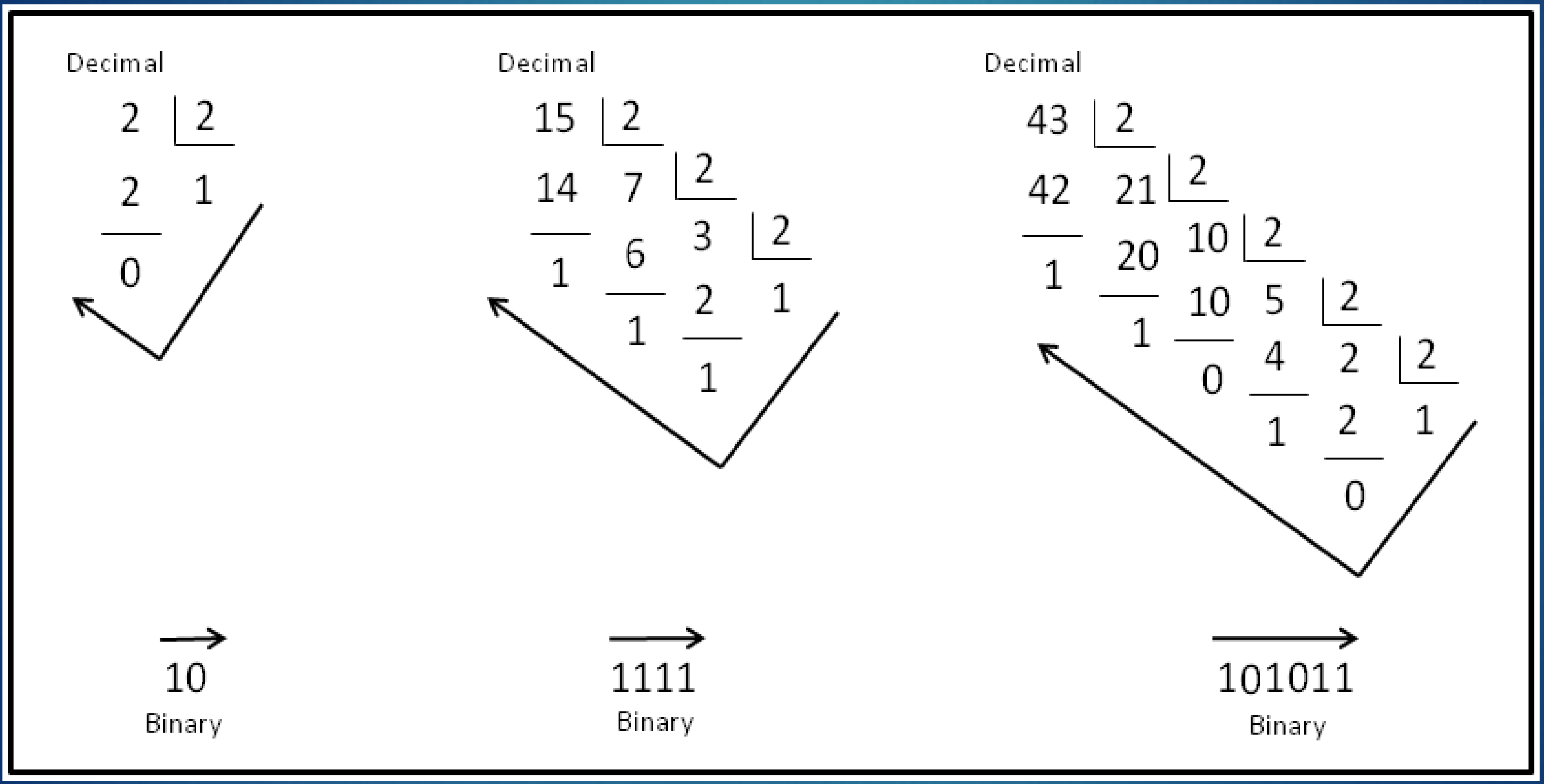
Figure 5.1: A recursion trace for the call factorial(5).

```
public static int factorial(int n) throws IllegalArgumentException {
  if (n < 0)
    throw new IllegalArgumentException( );    // argument must be nonnegative
  else if (n == 0)
    return 1;                                  // base case
  else
    return n * factorial(n−1);                 // recursive case
}
```

DECIMAL TO BINARY

BASIC RECURSIVE EXERCISES

DECIMAL TO BINARY EXAMPLE



ThS.Trần Lê Như Quỳnh

ThS.Trần Lê Như Quỳnh

ENGLISH RULER

BASIC RECURSIVE EXERCISES

DRAWING AN ENGLISH RULER

```
---- 0
-
--
-
---
-
--
-
---- 1
-
--
-
---
-
--
-
---- 2
```

(a)

```
----- 0
-
--
-
---
-
--
-
----
-
--
-
---
-
--
-
----- 1
```

(b)

```
--- 0
-
--
-
--- 1
-
--
-
--- 2
-
--
-
--- 3
```

(c)

Figure 5.2: Three sample outputs of an English ruler drawing: (a) a 2-inch ruler with major tick length 4; (b) a 1-inch ruler with major tick length 5; (c) a 3-inch ruler with major tick length 3.

DRAWING AN ENGLISH RULER


```
/** Draws an English ruler for the given number of inches and major tick length. */
public static void drawRuler(int nInches, int majorLength) {
  drawLine(majorLength, 0);                  // draw inch 0 line and label
  for (int j = 1; j <= nInches; j++) {
    drawInterval(majorLength − 1);           // draw interior ticks for inch
    drawLine(majorLength, j);                // draw inch j line and label
  }
}
private static void drawInterval(int centralLength) {
  if (centralLength >= 1) {                  // otherwise, do nothing
    drawInterval(centralLength − 1);         // recursively draw top interval
    drawLine(centralLength);                 // draw center tick line (without label)
    drawInterval(centralLength − 1);         // recursively draw bottom interval
  }
}
private static void drawLine(int tickLength, int tickLabel) {
  for (int j = 0; j < tickLength; j++)
    System.out.print("-");
  if (tickLabel >= 0)
    System.out.print(" " + tickLabel);
  System.out.print("\n");
}
/** Draws a line with the given tick length (but no label). */
private static void drawLine(int tickLength) {
  drawLine(tickLength, −1);
}
```

ThS.Trần Lê Như Quỳnh

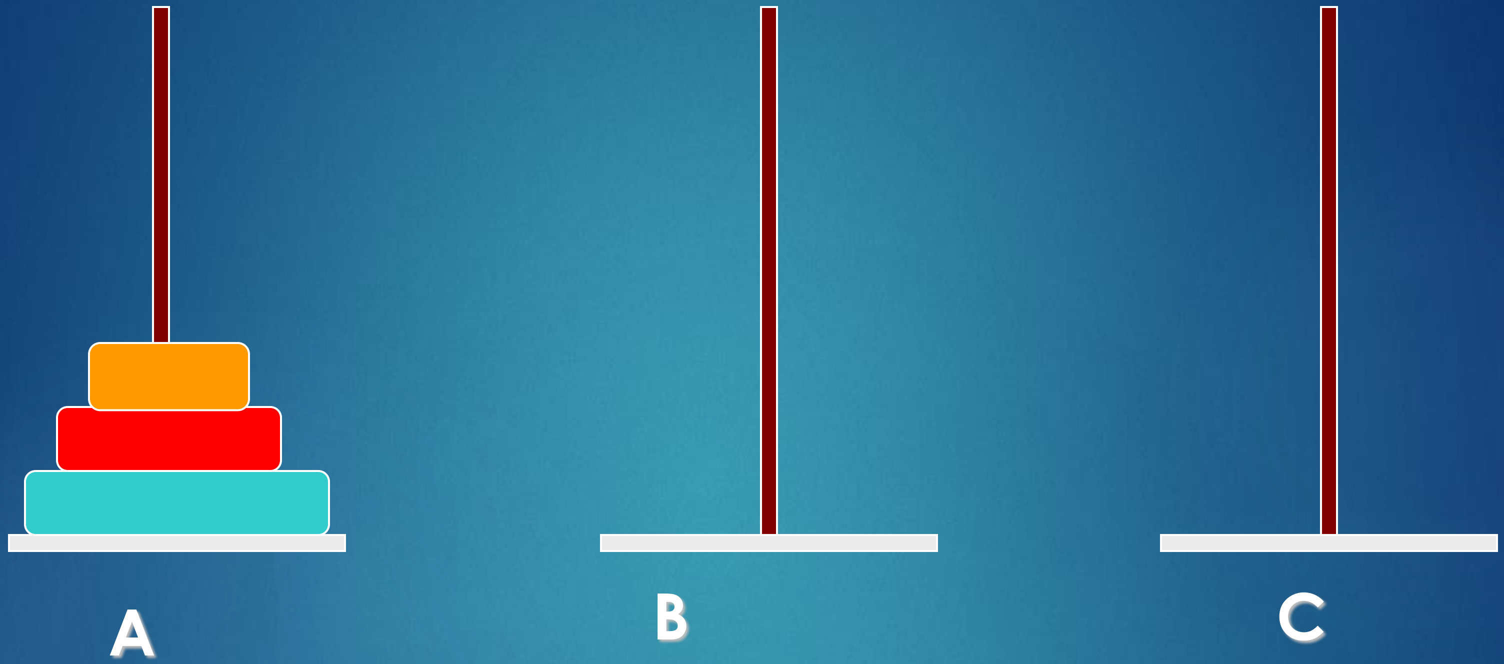
HA NOI TOWER

RULE OF HANOI TOWER GAME

ThS.Trần Lê Như Quỳnh

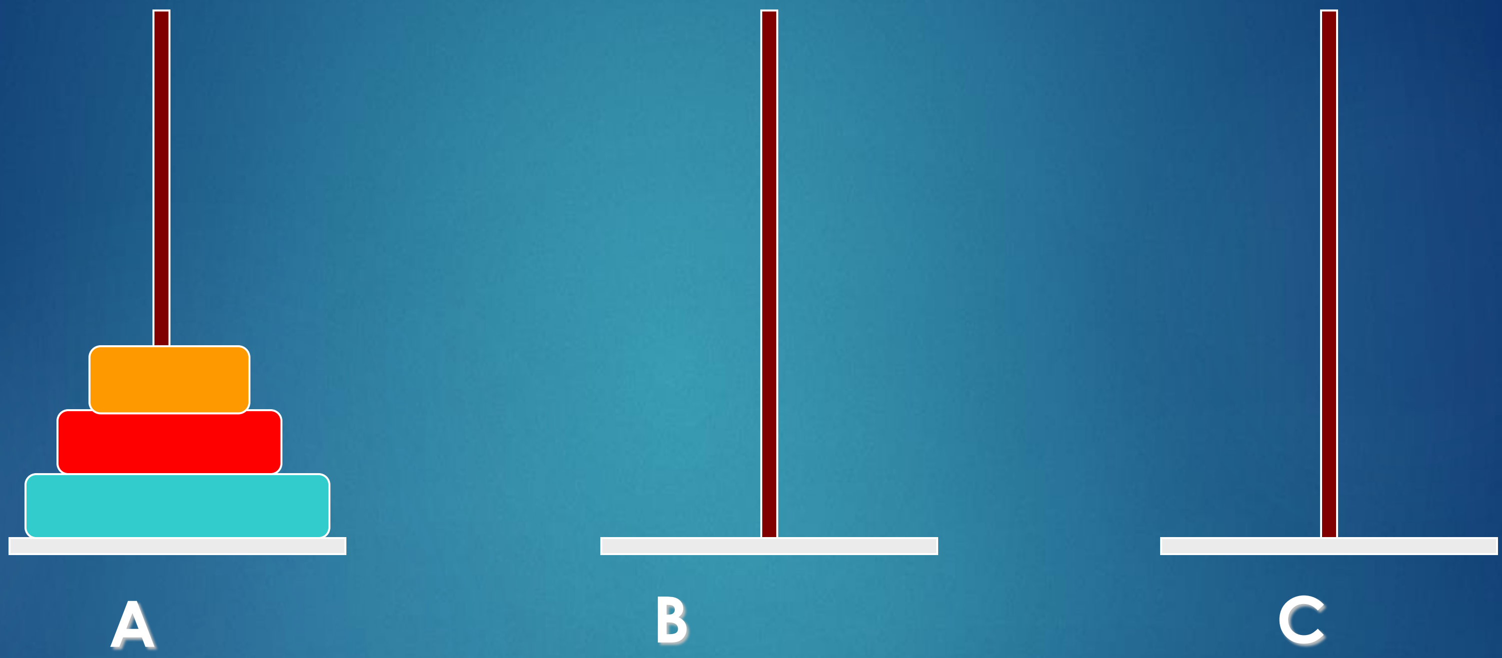
1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

HANOI TOWER GAME



How to move 3 disks from A column to C column with minimum of steps?

SOLUTION



- ***The minimum number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks.***

Recursive solution

ThS.Trần Lê Như Quỳnh

- RULE:
  - label 3 columns ***begin(A), middle(B), end(C)***
  - let n be the total number of discs
  - number the discs from 1 (smallest, topmost) to n (largest, bottommost)
  - To move n discs from ***begin*** column to ***end*** column
- RECURSIVE SOLVE:
  - Recursive move n−1 discs from begin to end. With **begin** not change, **end**, **middle *(swap end and middle column)***
  - move disc n from **begin** to **end**
  - Recursive move n−1 discs from begin to end. With **middle** , **begin**, **end** not change ***(swap begin and middle column)***

Tower of Hanoi Recursive Algorithm:

- **Function TowersofHanoi(n ,start, mid, end){**

//begin start A, mid B, end C

- N = number of disks

  If N == 1
  Move Single disk from A to C

- If   N >1
  Move n-1 disks from start A to
  B   TowersofHanoi(n-1,start, end , mid)

- Move last Disk from A to C

- Move n-1 disks from B to
  C.   TowersofHanoi(n-1,start, mid, end)

FIBONACCI

FIBONACCI IN NATURE

FIBONACCI IN NATURE

FIBONACCI

ThS.Trần Lê Như Quỳnh

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ \ldots$$

$$F(n) := \begin{cases} 1\,, & \text{khi } n = 1\,; \\ 1, & \text{khi } n = 2; \\ F(n-1) + F(n-2) & \text{khi } n > 2. \end{cases}$$

PASCAL'S TRIANGLE

ThS.Trần Lê Như Quỳnh

PASCAL'S TRIANGLE

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

1 10 45 120 210 252 210 120 45 10 1

$$(a+b)^0 = 1$$

$$(a+b)^1 = a+b$$

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

$$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$$

ThS.Trần Lê Như Quỳnh

PASCAL'S TRIANGLE

It is commonly called "n choose k" and written like this: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Notation: "n choose k" can also be written **C(n,k)**, $^{n}C_{k}$ or even $_{n}C_{k}$.

$$\binom{0}{0}$$

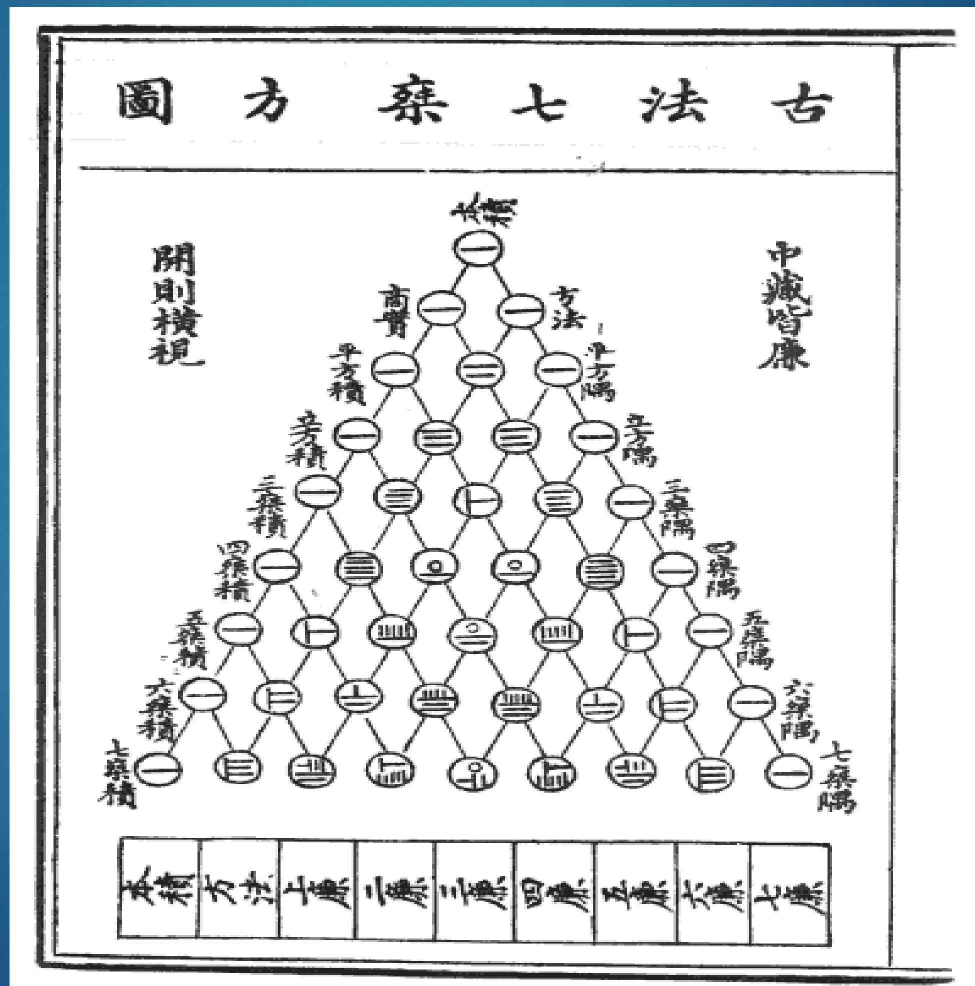
$$\binom{1}{0} \quad \binom{1}{1}$$

$$\binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2}$$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$

$$\binom{4}{0} \quad \binom{4}{1} \quad \binom{4}{2} \quad \binom{4}{3} \quad \binom{4}{4}$$

Chu Shi-Chieh's book "*Ssu Yuan Yü Chien*" - *1303*

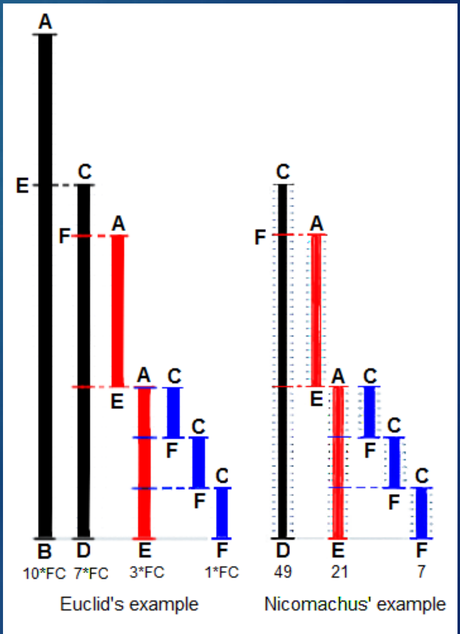


ThS.Trần Lê Như Quỳnh

ThS.Trần Lê Như Quỳnh

GREATEST COMMON DIVISOR (ước chung lớn nhất)

EUCLIDEAN ALGORITHM



Example

ThS.Trần Lê Như Quỳnh

- GCD(42, 56)=14

$$\frac{42}{56} = \frac{3 \cdot 14}{4 \cdot 14} = \frac{3}{4}.$$

Euclidean algorithm

- With a, b € N
- Condition a >= b, a, b ≠ 0
- Has a = b*q + r (with r = a mod b)

$$\mathrm{UCLN}(a,b) = \begin{cases} b & \text{nếu} \quad r = 0 \\ \mathrm{UCLN}(b,r) & \text{nếu} \quad r \neq 0 \end{cases}$$

ThS.Trần Lê Như Quỳnh

TRY TO SOLVE SOME ALGEBRA PROBLEMS

ThS.Trần Lê Như Quỳnh

ALGEBRA PROBLEMS

1. $S(n)=1-2+3-4+\ldots+((-1)^{(n+1)}).n, n>0)$
2. $S(n)=1+1.2+1.2.3+\ldots+1.2.3\ldots n, n>0$
3. $S(n)=1^2+2^2+3^2+\ldots.+n^2$, $n>0$
4. $S(n)=1+1/2+1/(2.4)+1/(2.4.6)+\ldots$ $+1/(2.4.6.2n), n>0$