# 4 Analysis of Recursive Functions

Thursday, February 13, 2025 12:59 PM

fib(n) n!

$F(n) = n! = (n-1)! \cdot n$

$n! = 1 * 2 * 3 * 4 * \ldots * n$

or

$$F(n) \begin{cases} F(n-1) \cdot n & n > 0 \quad \text{recursive case.} \\ 1 & n = 0 \quad \text{base case.} \end{cases}$$

E.G.

```
FUNCTION fact(n)
    IF n=0 THEN
        RETURN 1
    ELSE
        RETURN fact(n-1) * n
```

Analysis:

basic operation: *

$$M(n) = \underbrace{M(n-1)}_{\text{no. of multiplications inside Fact}(n-1)} + \underbrace{1}_{\text{fact}(n-1) * n}$$

"Recurrances" or "Recurrance Relations"

(Big in Discrete Math & Analysis of Algorithms

Note: it has infinite solutions.

e.g.

| n | ... | 9 | 10 | 11 | 12 | 13 | 15 | .... |
|---|---|---|---|---|---|---|---|---|
| M(n) | | 7 | 8 | 9 | 10 | 11 | 12 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $M(n)$ | | 7 | 8 | 9 | 10 | 11 | 12 |

$M(10) = M(9) + 1$
$M(11) = M(10) + 1$

We also need an initial condition.

$$M(0) = 0$$

So:

- $M(n) = M(n-1) + 1$
- $M(0) = 0$

- BACKWARDS SUBTITUTION

$M(n) = M(n-1) + 1$

- $M(n-1) = M(n-2) + 1$

$= M(n-2) + 1 + 1$

- $M(n-2) = M(n-3) + 1$

$= M(n-3) + 1 + 1 + 1$

- $M(n-3) = M(n-4) + 1$

$= M(n-4) + 1 + 1 + 1 + 1$

after $i$ of substitutions.

$M(n) = M(n-i) + i$

Let $i = n$

$= M(n-n) + n$

$= M(0) + n$

$= 0 + n$

$M(n) = n \in \Theta(n)$ Linear.

GENERAL PLAN FOR ANALYSING THE EFFICIENCY OF <u>RECURSIVE</u> ALGORITHMS

1) Decide on parameter(s) that indicate input size.

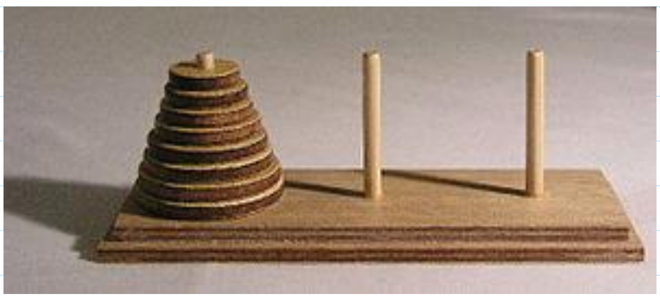
2) Identify basic operation.

3) Check whether the number of times the basic operation is performed depends only on the input size.

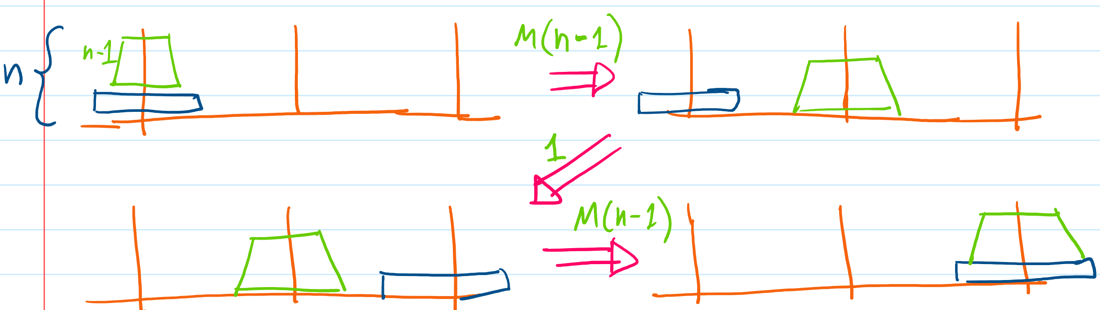
4) Set up a recurrence relation with its initial condition.

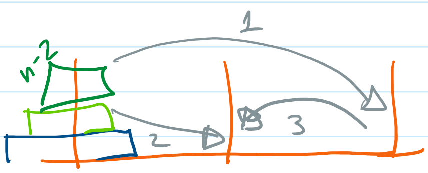
5) Solve the recurrence. ( using known tools )
or at least, find the order-of-growth of the solution.

E.G. the towers of Hanoi

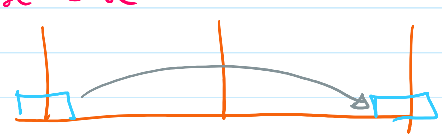


https://www.mathsisfun.com/games/towerofhanoi.html




Base case $n=1$



Let's count how many moves are needed to move n discs.

- $M(n) = M(n-1) + 1 + M(n-1)$ recurrence
- $M(1) = 1$ initial condition

- Solve by backwards substitution.

$$M(n) = 2 \cdot M(n-1) + 1$$

$$M(n-1) = 2 \cdot M(n-2) + 1$$

$$M(n) = 2 \cdot (2 \cdot M(n-2) + 1) + 1$$
$$= 2^2 \cdot M(n-2) + 2 + 1$$

$$M(n-2) = 2 \cdot M(n-3) + 1$$

$$= 2^2 \cdot (2 \cdot M(n-3) + 1) + 2 + 1$$
$$= 2^3 \cdot M(n-3) + 2^2 + 2 + 1$$

$$M(n-3) = 2 \cdot M(n-4) + 1$$

$$= 2^3 \cdot (2 \cdot M(n-4) + 1) + 2^2 + 2 + 1$$
$$= 2^4 \cdot M(n-4) + 2^3 + 2^2 + 2 + 1$$

after $i$ substitutions

$$= 2^i \cdot M(n-i) + 2^{i-1} + 2^{i-2} + 2^{i-3} + \dots 2^3 + 2^2 + 2^1 + 2^0$$
$$= 2^i \cdot M(n-i) + \sum_{k=0}^{k=i-1} 2^k$$
$$= 2^i \cdot M(n-i) + 2^i - 1$$

Let $i = n-1$

$$M(n) = 2^{n-1} \cdot \underbrace{M(n-(n-1))}_{} + 2^{n-1} - 1$$
$$= 2^{n-1} \cdot 1 + 2^{n-1} - 1$$
$$= 2 \cdot 2^{n-1} - 1$$
$$M(n) = 2^n - 1 \in \Theta(2^n)$$ exponential.

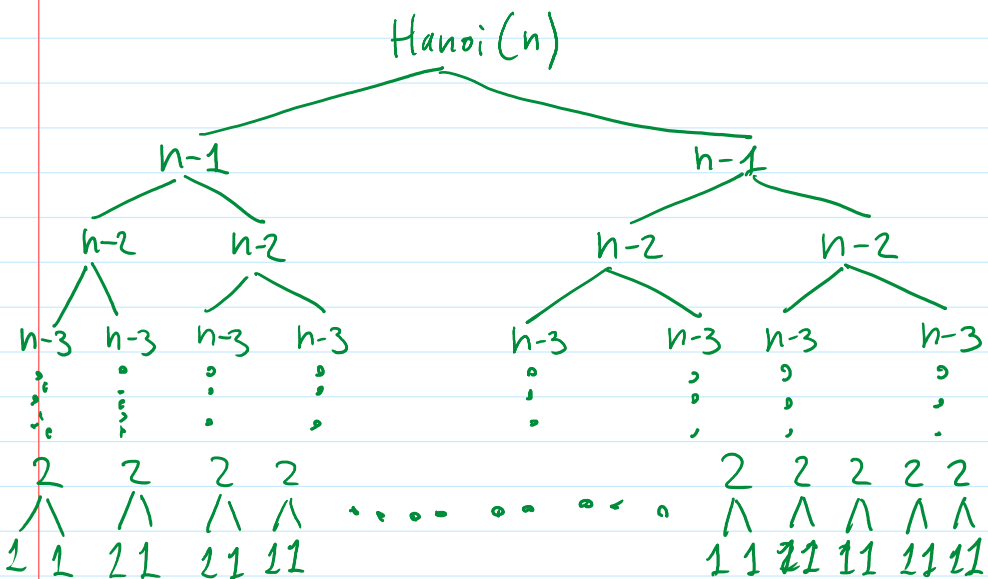
$$M(9) = 511$$
$$M(10) = 1023$$
$$\vdots$$
$$M(64) = 18{,}446{,}744{,}073{,}709{,}551{,}615$$

- Alternative Method : "Using the "call" tree"

• Alternative Method: Using the Call Tree

FUNCTION Hanoi(n)
  Move n-1 disks : Hanoi(n-1)
  Move largest disk
  Move n-1 disks : Hanoi(n-1)

Function call tree:




$1 = 2^0$

$2 = 2^1$

$4 = 2^2$

$8 = 2^3$

$\vdots$

$2^{n-2}$

$2^{n-1}$

$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-1}$

$$2^n - 1 = \sum_{i=0}^{n-1} 2^i$$

$2^n - 1 \in \Theta(2^n)$ exponential.

# E.G. BinLenRec.

FUNCTION BinLenRec.(n)
  // the number of binary digits needed to represent n
  IF n=1 OR n=0
    RETURN 1
  ELSE
    RETURN BinLenRec($\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$) + 1

Analysis:
  basic operation +

• $A(n) = 1 + A(\lfloor \frac{n}{2} \rfloor)$   Recurrence

- $A(n) = \underbrace{1}_{\text{add one}} + \underbrace{A(\lfloor \frac{n}{2} \rfloor)}_{\text{additions needed by recursive call}}$ Recurrence

- $A(1) = 0$ initial condition.

- Tools: Let $n = 2^k$ ; $k = \log_2 n$

$\boxed{\text{by the "smoothness rule" we can safely make this substitution}}$

- $A(2^k) = 1 + A\left(\frac{2^k}{2}\right)$
- $A(1) = 0$

Solve by backwards substitution.

$$A(2^k) = A(2^{k-1}) + 1$$

$$A(2^{k-1}) = A(2^{k-2}) + 1$$

$$= A(2^{k-2}) + 1 + 1$$

$$A(2^{k-2}) = A(2^{k-3}) + 1$$

$$= A(2^{k-3}) + 1 + 1 + 1$$

$$A(2^{k-3}) = A(2^{k-4}) + 1$$

$$= A(2^{k-4}) + 1 + 1 + 1 + 1$$

After $i$ substitutions

$$A(2^k) = A(2^{k-i}) + i$$

Let $i = k$ to get to initial conditions

$$A(2^k) = A(2^{k-k}) + k$$

$$= A(2^0) + k$$

$$= A(1) + k$$

$$A(2^k) = k$$

So

$n = 2^k$ $k = \log_2 n$

$$A(n) = \log_2 n \in \Theta(\log n)$$

—o— EOF