4 Analysis of Recursive Algorithms

Tuesday, September 10, 2024 11:09 AM

$n!$ $\quad fib(n)$

- factorial

$$F(n) = n! = 1 \cdot 2 \cdot 3 \cdot 4 \ldots \cdot (n-1) \cdot n$$
$$= (n-1)! \cdot n$$

or

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

E.G.

```
FUNCTION fact(n)
    if n=0
        return 1
    else
        return fact(n-1) * n
```

Analysis:

basic_operation.

$$M(n) = \underbrace{M(n-1)}_{\text{no. of multiplications to compute } F(n-1)} + \underbrace{1}_{\text{multiply } F(n-1) \text{ \& } n}$$

"Recurrances" or "Recurrance Relations"
(Big in discrete math and Analysis of Algorithms)

Note: it has infinite solutions.

n	9	10	11	12	13	14	15	16		
$M(n)$	12	13	14	15	16	17	18	19		

$M(13) = M(12) + 1$
$M(14) = M(13) + 1$

But. we also have an initial condition.

$M(0) = 0$

collecting

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

- BACKWARD SUBSTITUTION

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

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

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

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

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

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

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

after i subtitutions.

$= M(n-i) + i$

Let $i = n$

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

$= M(0) + n$

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

- GENERAL PLAN FOR ANALYSING THE TIME EFFICIENCY OF RECURSIVE ALGORITHMS

1.- Decide on the parameters for the size of the input

2.- Identify basic_operation

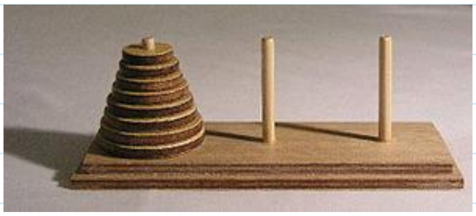
3.- Check whether the number of execution of the basic operation depends solely on the size of the input, or on other factors.

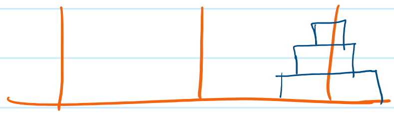
4.- Set up a recurrance relation. with it's initial condition for the number of times the basic operation is executed.

with it's initial condition
for the number of times the basic operation is executed.

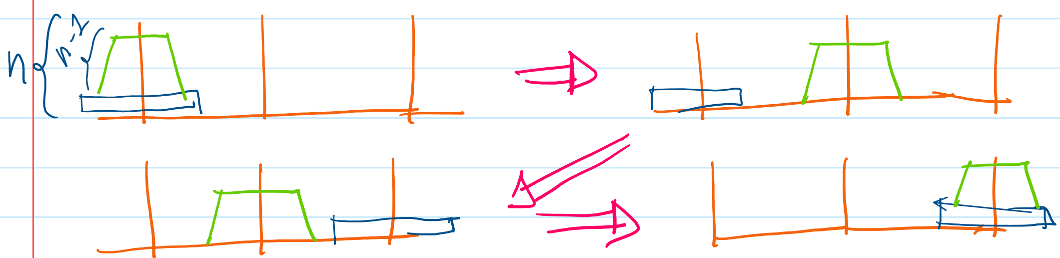
5.- Solve the recurrance. (using known results and tools)
or at least, find the order-of-growth of the solution.

E.G 2 Towers of hanoi





n=3.



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

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

SUMS : REVIEW #2

- $\sum_{i=0}^{u} a^i = \dfrac{a^{u+1} - 1}{a-1}$
  $= a^0 + a^1 + a^2 + \ldots + a^u$
- $\sum_{i=0}^{u} 2^i = 2^{u+1} - 1$
  $= 2^0 + 2^1 + 2^2 + \ldots 2^u$

Solve by backwards subtitution:

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

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

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

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

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

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

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

after i subtitutions ...

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

Let $i = n-1$ to reach initial condition

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