Content

- Recurrence Relation → Complexity analysis
- → Efficiently solve recurrence relation (Matrix Expo)
- → Bit manipulation → DP with Bitmask
- → Combinatorics (P, C)
- → Probability → DP with probability, Expectation
- → Pigeonhole principle
- →
- →

Pre-requisite → Programming constructs

Recursion

<u>NT</u>

Complexity Analysis

Computer system

→ RAM

→ CPU

→ HDD/SSD

⋮

→

→ we can work efficiently on the machine

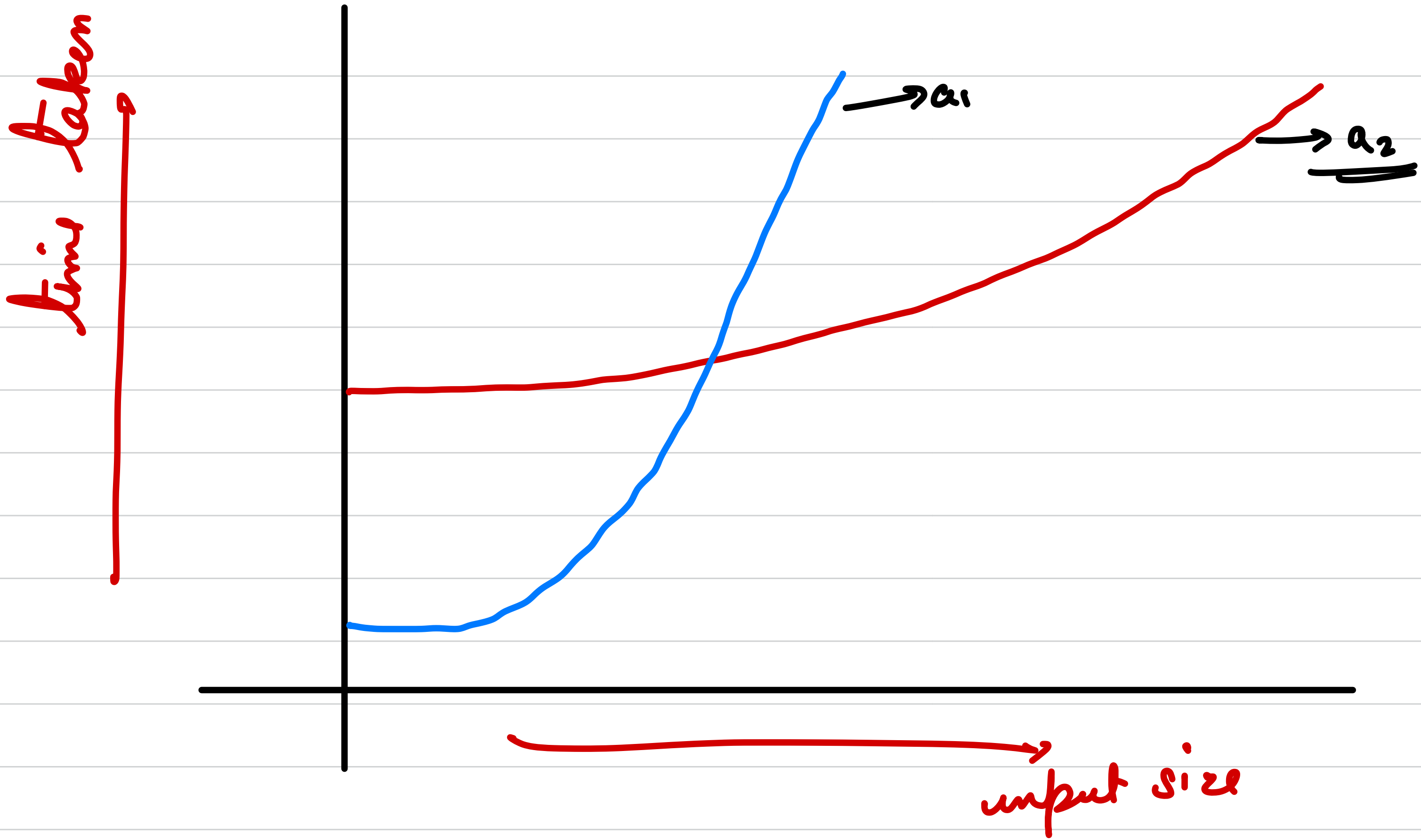
Software → efficient → Time

→ Space

1 sec → approx 10^8 — 10^9

→ **algorithm** → How can we time taken by the **algorithm ??**

- execution time
- No. of statements executed
- how the algorithm performs on a input and it's growth rate.



time taken
input size
a1
a2

Rate of growth ?? → Rate at which the running time increases as a func of input

$y = f(x)$

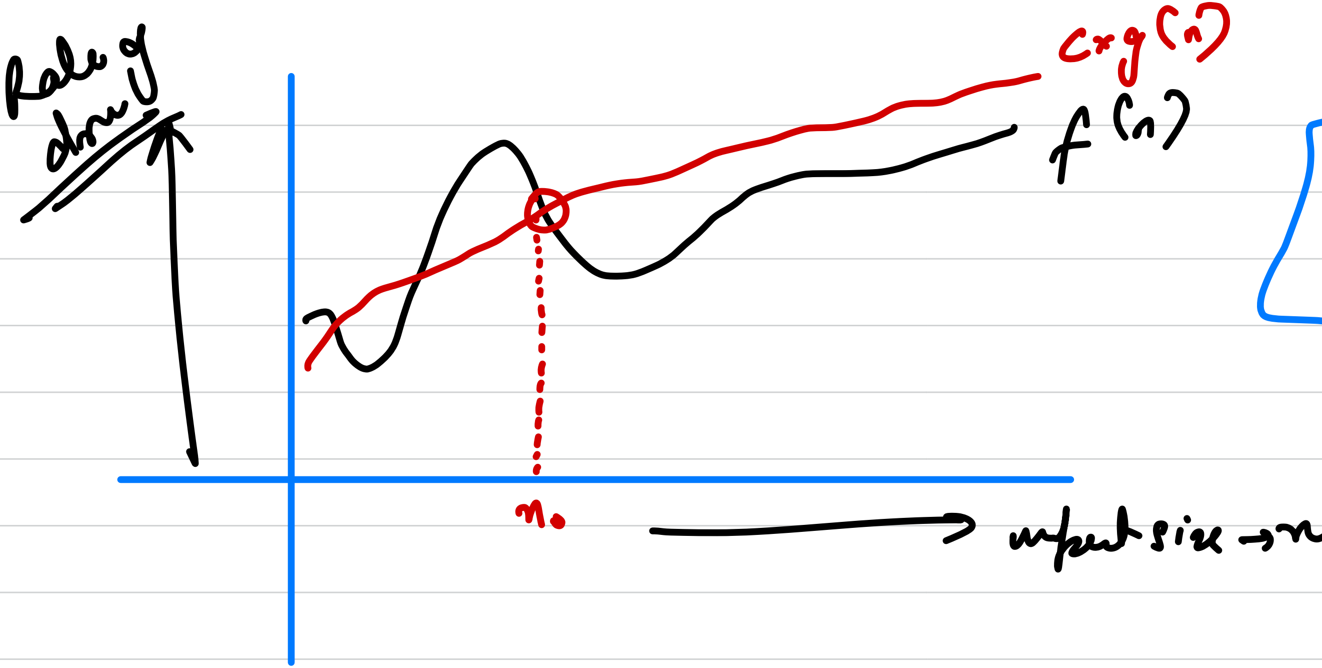
function take input size as argum

Best Case ✓

Avg Case ✓

Worst Case ✓

→ asymptotic analysis



Big O notation

$C \rightarrow$ constant

Big O notation $\Rightarrow$ gives the tightest upper bound of a function

$$\longrightarrow f(n) = O(g(n)) \longleftarrow$$

$$f(n) = n^4 + 99n^2 + 35n + 6$$

$g(n) \Rightarrow$ gives the max rate of growth for $f(n)$ at large values of n

$$\underline{\underline{g(n) = n^4}}$$

$O(g(n)) = \{ f(n) :$ there exist +ve constant

c and n_0 such that

$$0 \le f(n) \le c\,g(n) \;\; \forall \; n \ge n_0 \}$$

↳ tightest upper bound

$f(n) = n^2 + 20$

$g(n) = n^2$

$c = 2$

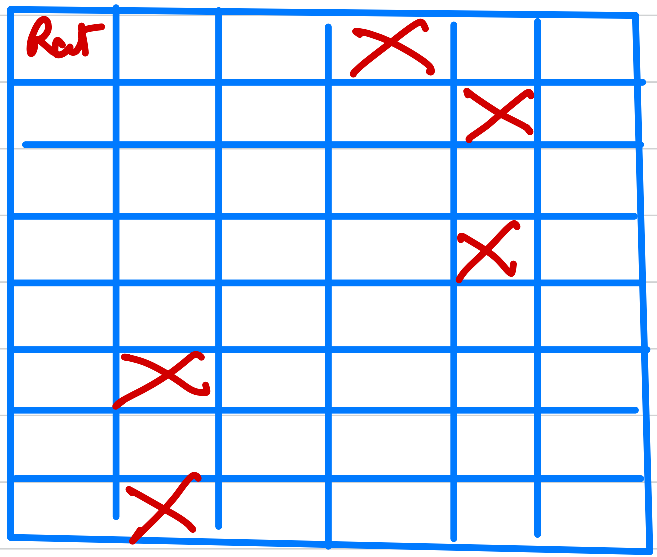
$f(100) = (100)^2 + 20 \rightarrow \boxed{10020}$

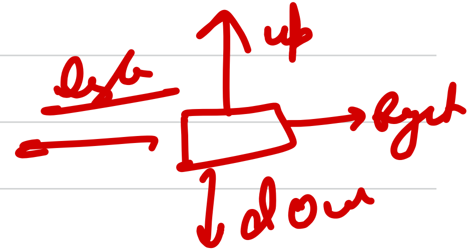
$c\,g(100) = 2 \times (100)^2$

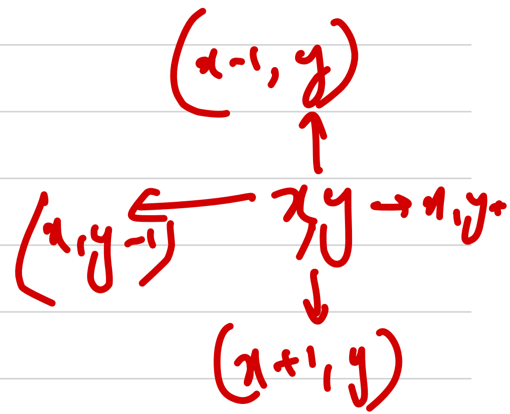
↳ $\boxed{20000}$

Example → Rat in a maze

Total ways such that Rat reaches from TL to BR







Time Complexity → Tightest upper Bound

$$f(x,y) = f(x-1,y) + f(x+1,y) + f(x,y-1) + f(x,y+1)$$

return the no. of ways in which you can reach from (x,y) to BR

$4n^2$ ← good upper bound

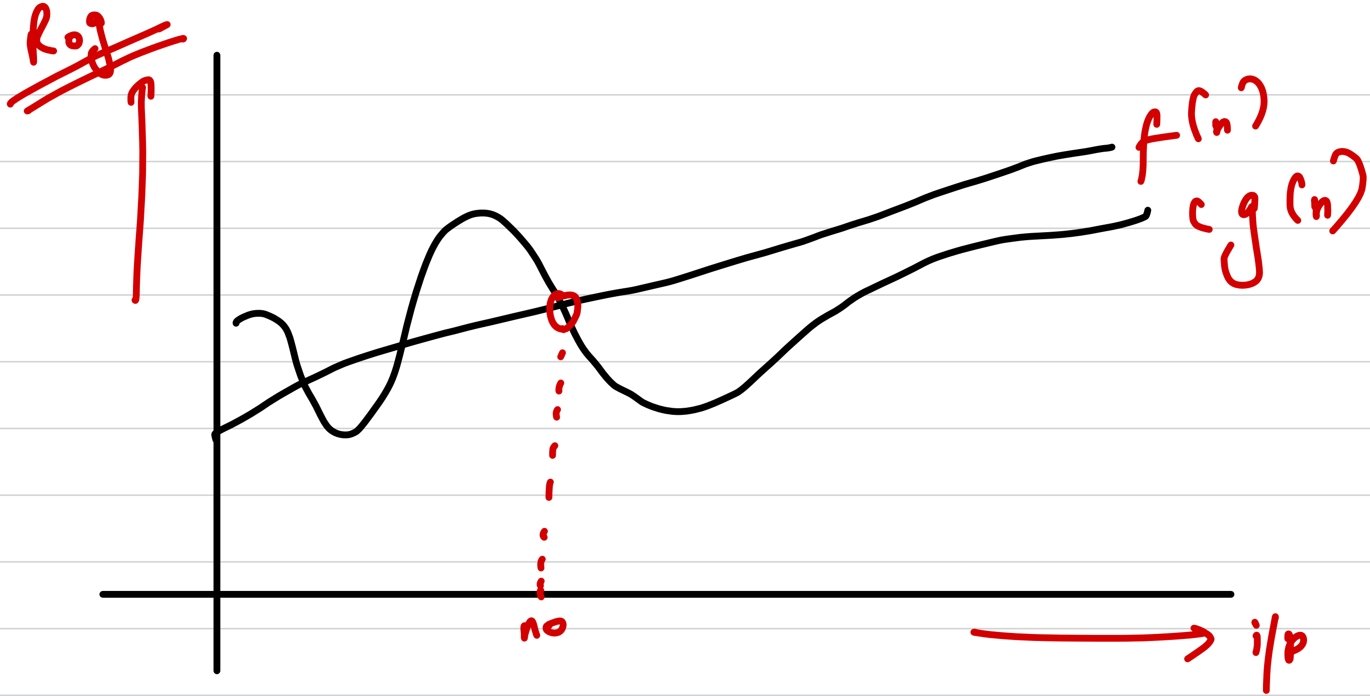
$3n^2$

$2^{(\text{no. of blank cells})} \times$

nothing is possible < tightest bound < everything is possible

so a loose bound

little o notation



Tightest lower bound

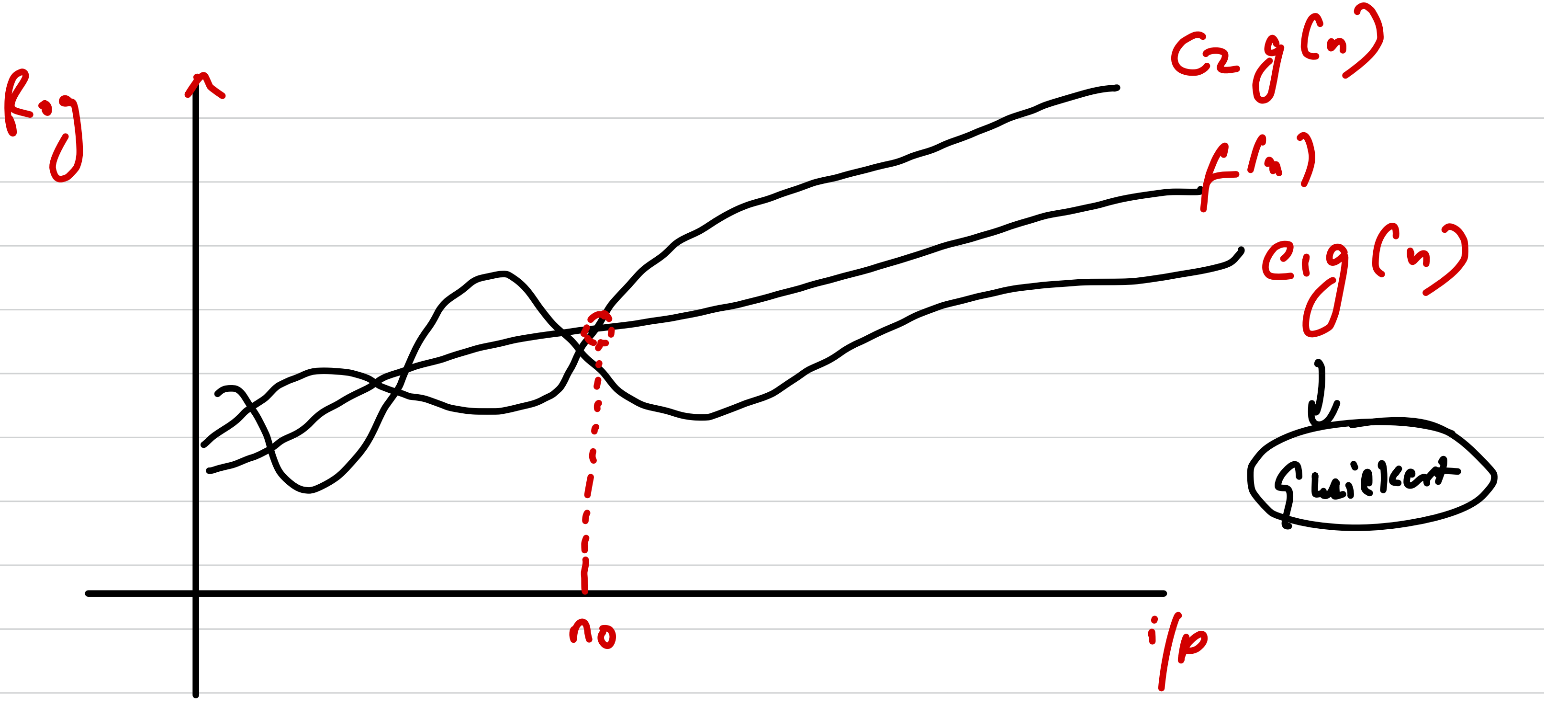
$f(n) = \Omega(g(n))$

$0 \leq cg(n) \leq f(n)$

$f(n) = 100n^2 + 10n + 60$

$g(n) = n^2$

$C = 100$



average case (Θ) (theta)

$c_1 g(n) \le f(n) \le c_2 g(n)$

$f(n) = \Theta(g(n))$

$\forall\ n \ge n_0$

$\rightarrow$ iteration

$\rightarrow$ recursion

$$\boxed{f_n = f_{n-1} + f_{n-2}}$$

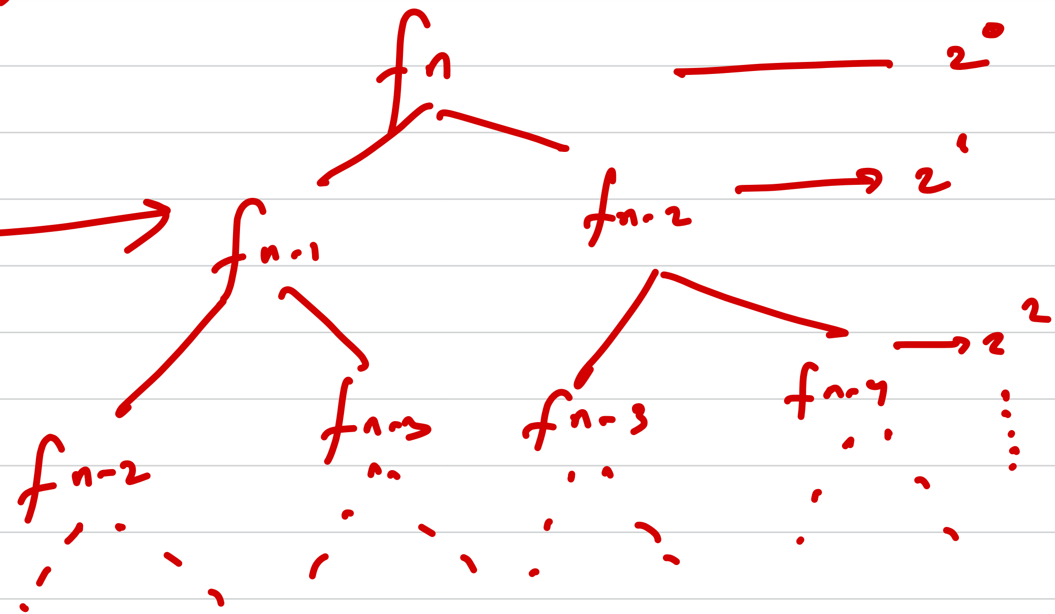
$$\boxed{f_n = n \times f_{n-1}}$$

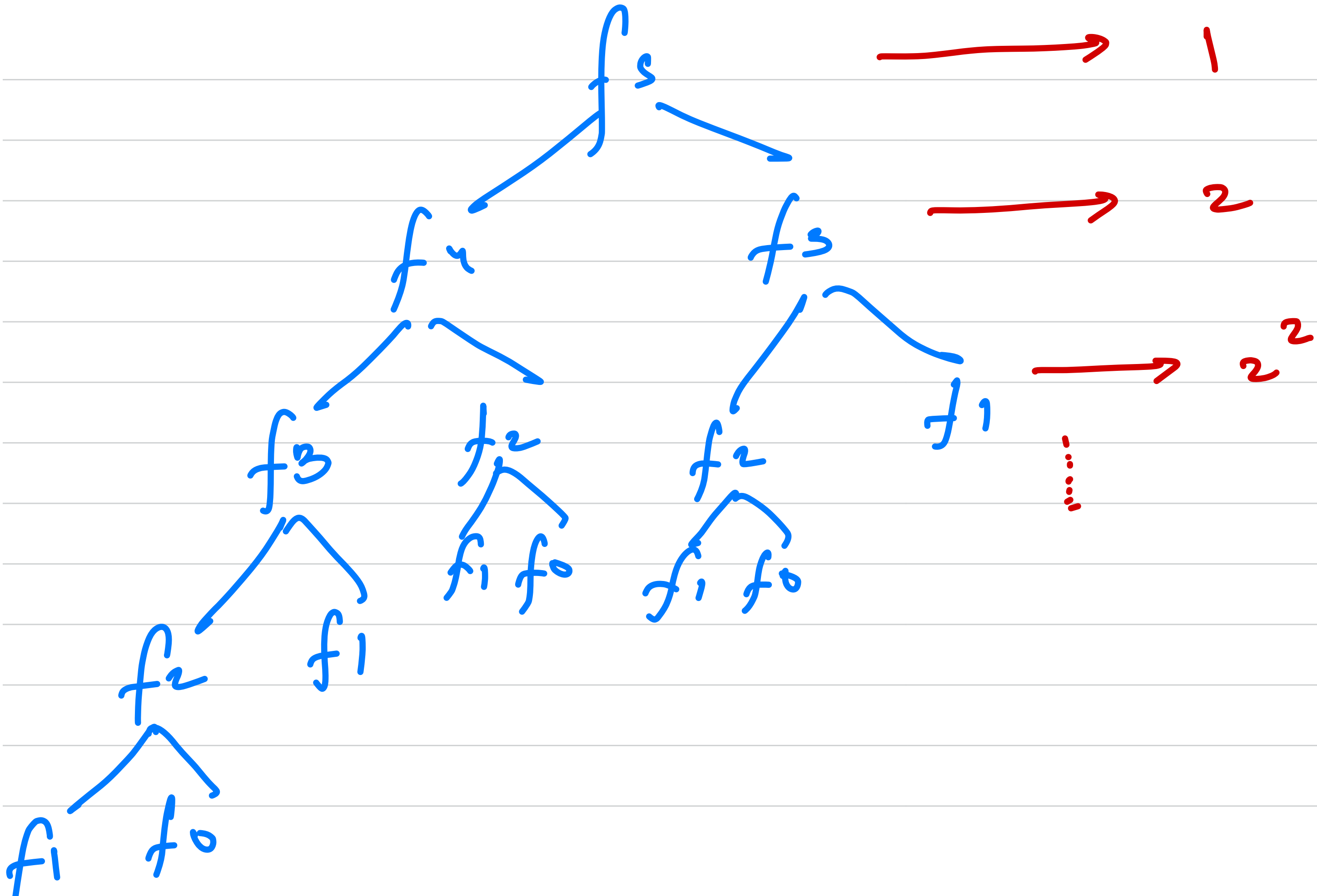
Recursive

fibonacci → $f(n) = \underbrace{f(n-1) + f(n-2)}$

highest upper bound for the func

Recursion Tree →





f5
f4
f3
f3
f2
f2
f1
f2
f1
f1
f0
f1
f0
f1
f0
1
2
2²

$$\left(2^0 + 2^1 + 2^2 \cdots\cdots 2^{n-2}\right) \rightarrow \frac{1\times(2^{n-1}-1)}{}$$

$x+y$

GP $\rightarrow$ $r \rightarrow$ ratio

$r>1$ $\quad r<1$

a - constant

$$a \quad ar \quad ar^2 \cdots\cdots ar^{n-1}$$

$\rightarrow$ last term

Sum $\rightarrow a\left(1, r, r^2 \cdots\cdots r^{n-1}\right)$ $\rightarrow$ total term

$$a \times \frac{1\times(r^n-1)}{r-1} \Rightarrow \frac{a(r^n-1)}{r-1} \quad r>1$$

$$\frac{a(1-r^n)}{1-r} \quad r<1$$

$$(2^n - 1 + x + y) \longrightarrow c \times 2^n \longrightarrow \underline{\underline{O(2^n)}}$$

$C \to$ const

$\Rightarrow$ $$f(n) = \underbrace{2f\left(\frac{n}{2}\right) + c*n}$$

$$f(n) = 3f\left(\frac{n}{2}\right) + cn^2$$

$$\Downarrow$$

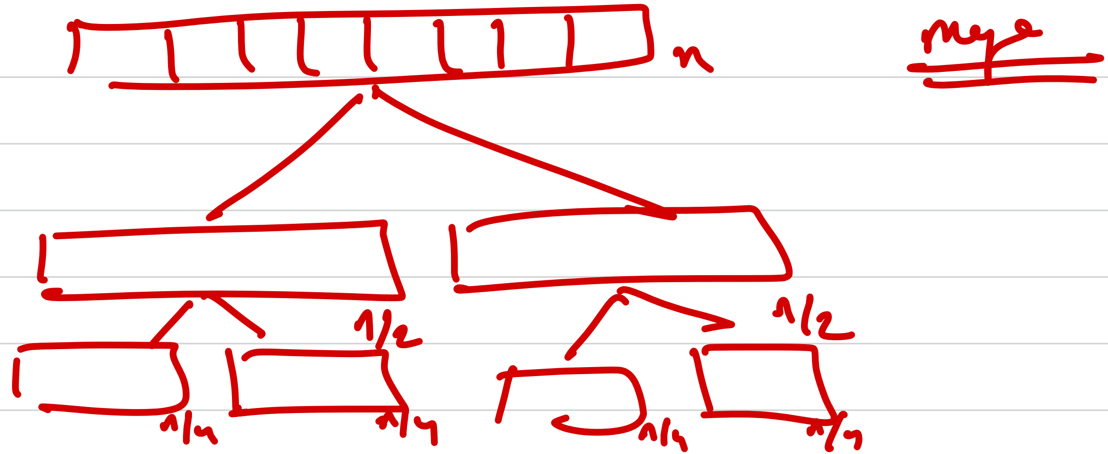
$$f(n) = a\,f\left(\frac{n}{b}\right) + h(n)$$

general eqⁿ of D n C

$a \to$ no. of subproblem

$\left(\frac{n}{b}\right) \to$ size of new subproblem

$h(n) \to$ polynomial funⁿ of n

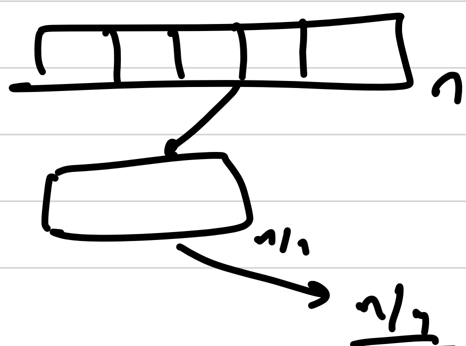


$$ms(n) = ms(n/2) + ms\left(\frac{n}{2}\right) + merge\left(\frac{n}{2}, \frac{n}{2}\right)$$

$$f(n) = f\left(\frac{n}{2}\right) + c$$

$a = 1$ $\quad$ $h(n) = c\,n^0$

$\frac{n}{b} \to \frac{n}{2}$

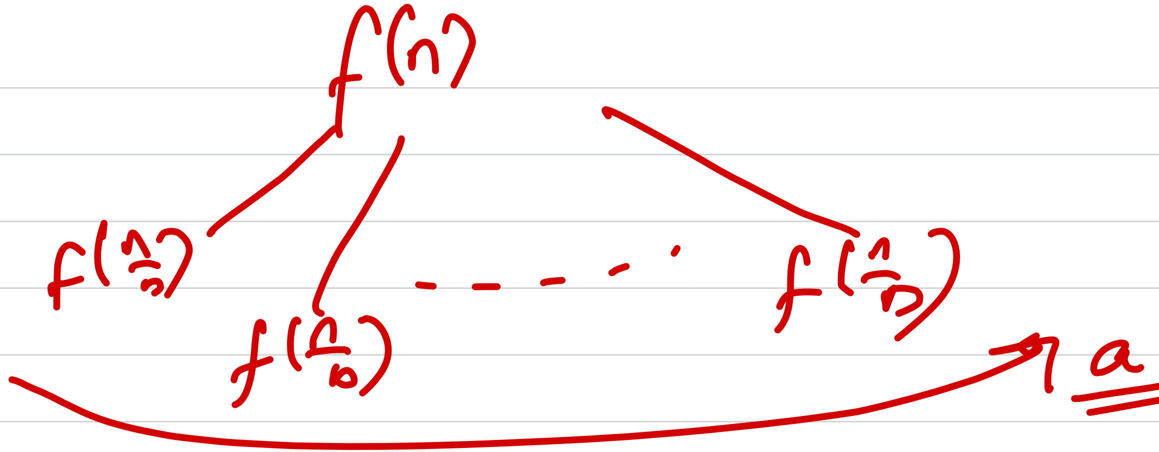


$$f(n) = a f\left(\frac{n}{b}\right) + h(n)$$

→ genal eqⁿ g
DnC

multiple approach

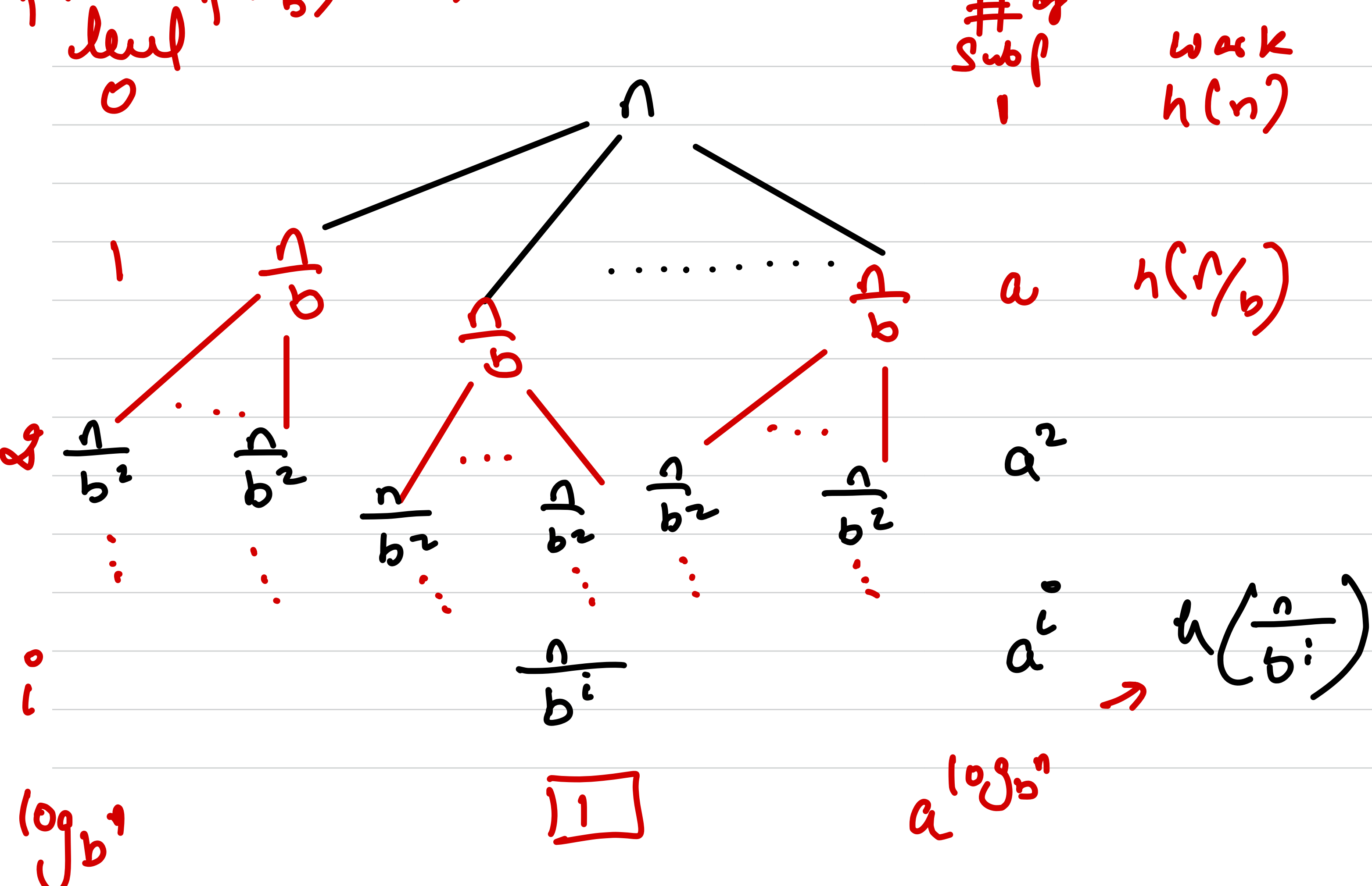
- → Recursn Tree
- → Master Theorem
- → akra bazi formula
- → assumption
- → substitn



Master theorem ← formula

demo ← Trep

$f(n) = af\left(\frac{n}{b}\right) + h(n)$



a^i

$n \rightarrow \frac{n}{b} \rightarrow \frac{n}{b^2} \text{ - - - - - - - - } \frac{n}{b^k}$

$\frac{n}{b^k} = 1$

$n = b^k$

$\boxed{k = \log_b n}$

$$a^i \times h\left(\frac{n}{b^i}\right)$$

$h(n) \rightarrow$ polynomial on n

$h(n) = n^k$

$$h\left(\frac{n}{b^i}\right) = O\left(\frac{n}{b^i}\right)^k$$

$$a^i \times O\left(\frac{n}{b^i}\right)^k \Rightarrow \boxed{O(n^k)\left(\frac{a}{b^k}\right)^i}$$

$$\text{Total} \Rightarrow \sum_{i=0}^{\log_b n} O(n^k)\left(\frac{a}{b^k}\right)^i \rightarrow \text{gp}$$

if $\frac{a}{b^k} < 1$ i.e $r < 1$

$$a,\ ar,\ ar^2 \ \text{- - - - - - - -}\ ar^{n-1}$$

$\underline{r < 1}$ $\rightarrow$ $ar\left(\frac{1-r^n}{1-r}\right) \rightarrow O(a)$

$r > 1 \rightarrow \frac{a(r^n - 1)}{r-1} \rightarrow O(ar^n)$

$r == 1 \rightarrow \underline{O(na)}$

$\frac{a}{b^k} < 1 \longrightarrow \underline{\underline{O(n^k)}}$

Bloomberg

$\frac{a}{b^k} > 1 \longrightarrow O\left(O(n^k)\left(\frac{a}{b^k}\right)^{\log_b n}\right)$

$\boxed{x^{\log_y n} \to n^{\log_y x} \qquad x^{\log_x n} \Rightarrow \underline{\underline{n}}}$

$\longrightarrow O\left(O(n^k)\frac{a^{\log_b n}}{b^{k \log_b n}}\right)$

$\longrightarrow O\left(O(\cancel{n^k})\frac{n^{\log_b a}}{\cancel{n^k}}\right)$

$\longrightarrow O(n^{\log_b a})$

$\frac{a}{b^k} = 1 \rightarrow O\left(\left(\log_b n + 1\right) \times O\left(n^k\right)\right)$

$\rightarrow O\left(n^k \log_b n\right)$

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

$a = 3$ $\qquad$ $b = 4$ $\qquad$ $n^{k} =$ $k = 1$

$$\frac{a}{b^k} \rightarrow \frac{3}{4} < 1$$

$$O(n\log n)$$

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

$a = 4$ $\qquad k = 2$

$b = 2$

$$\frac{a}{b^k} \Rightarrow \frac{4}{2^2} \Rightarrow \frac{\cancel{4}}{\cancel{4}} = 1$$

$$O(n^2 \log_2 n)$$

$$O(n^2 \log n)$$