MADE EASY
COMPUTER SCIENCE
Topper Handwritten Notes
ALGORITHM
BY-RAVI SIR

- Theory
- Explanation
- Derivation
- Example
- Shortcuts
- Previous Years Question With Solution

ALGORITHM

-6 [
1. Notations
2. TC/SC of loops & Recursive Algo
3. Methods to solve Rec. Relations (Substitution, Rec. Tree, Master Methods)

6 [
4. Algo design techniques
   - divide & conquer
   - Greedy
   - DP

2 [5. Binary Heaps

2 [
6. Graph Traversal
7. Sorting algo

INTRODUCTION

Algorithm : step by step rep of computer program.

Finite Number of steps to perform some task.

Criteria of Algorithm

① finiteness – algo must terminate in finite amt of time.

② Definiteness – each step of algo must be unambiguous (have a unique sol^n)

(Such algo are also called Deterministic Algo) Analogy: DFA

Non Deterministic Algo: for each step of algo, there exists finite no. of solutions. Analogy: NFA

→ Algo should choose correct sol^n in one attempt

→ Not possible to run in computers.

eg a[1...n] searching

Det Algo	Non Det Algo
#comp = n	#comp = 1
feasible	Not feasible

Det Algo:

```
for(i=1;i<=n; i++)
{
    if (x==a[i])
        return i;
    else
        return -1;
}
```

Non Det Algo:

```
i = Choose (1,n);
if (x==a[i])
    return i;
else
    return -1;
```

#comp = n (Det Algo), feasible

#comp = 1 (Non Det Algo), Not feasible

Steps to design Algo

1) **Devise an Algo**

Design algo for given prob by using best design technique.

Problem (P) →
- Divide & Conquer
- greedy
- DP
- Brute force
- Back Tracking
- Branch & Bound
- etc..

) **Validation of Algo**

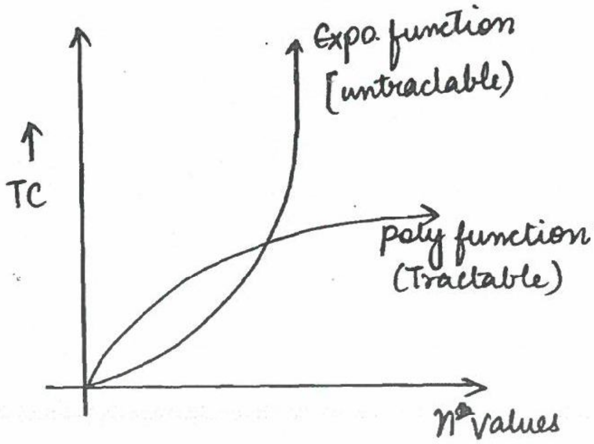
Test the result of algo is correct or not.

) **Analysis of Algo**

Estimate CPU time, main memory space reqd to complete the execution of algo.

Implement Algo and test Computer Program

Decidable Problem	Undecidable Problem
1. Problem for which, efficient algo exists. [Problem which can be solved in polynomial time by using Deter. algo] finite i/p → Decidable Problem → Halt in finite time 2. for n i/p size: #CPU comp: $[n, n\log n, n^2, n^3 ..]$ //Poly time	1. Problem for which no efficient algo exists. [Problem which req exponential time by using det. algo.] finite i/p → Undecidable Problem → Halt May not term. in finite time 2. for n i/p size: #CPU comp: $[2^n, 3^n, n!, n^n ...]$ //Exponent time

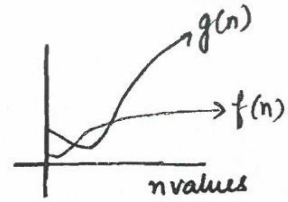


ASYMPTOTIC NOTATIONS

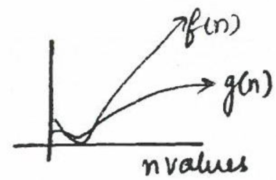
Asymptotic Comparison:

Asymp. comparison of non negative functions $f(n), g(n)$ is the growth rate comparison for large n values ($n \to \infty$).

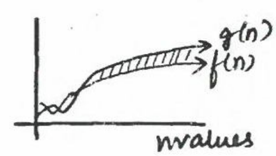
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$ iff $g(n)$ Asymptotically bigger than $f(n)$



$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$ iff $g(n)$ Asymptotically smaller then $f(n)$



$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \text{const}$$ iff $g(n)$ & $f(n)$ are asymptotically equal



① $f(n) = 1000n^2 + 100n + 50$

$g(n) = n^3 + 20n + 5$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n^2\left(1000 + \frac{100}{n} + \frac{50}{n^2}\right)}{n^2\left(n + \frac{20}{n} + \frac{5}{n^2}\right)} = \frac{1000}{\infty} = 0$$ $\therefore$ $g(n)$ asymp. bigger than $f(n)$

② $f(n) = 50n^2 + 20$

$g(n) = 100n + 100$

$$\lim f(n) = n^2\left(50 + \frac{20}{n^2}\right) = \frac{\infty}{100} = \infty$$ $\therefore$ $g(n)$ is asymp smaller than $f(n)$

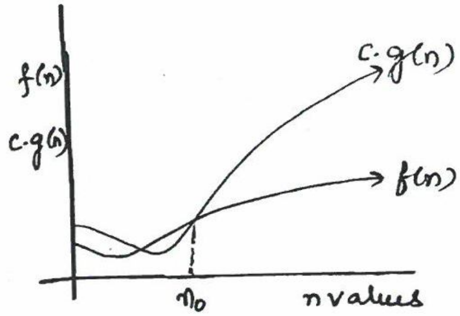
BIG O'h NOTATION

$f(n), g(n)$ non negative functions

$f(n) = O(g(n))$ iff $f(n) \leq c.g(n)$

for $\forall n$ where $n \geq n_0$

[c, n_0 are +ve const]



Q $f(n) = 10n+5$, $g(n) = n$

$$\left.\begin{array}{l} f(n) = 10n+5 \\ c.g(n) = 11.n \end{array}\right] \quad 10n+5 \leq 11.n \text{ for } n \geq 5$$

$\therefore f(n) = O(g(n))$

Q $f(n) = 10n+5$, $g(n) = n^2$

$$\left.\begin{array}{l} f(n) = 10n+5 \\ c.g(n) = 1.n^2 \end{array}\right] \quad 10n+5 \leq 1.n^2 \text{ for } n \geq 11$$

$\therefore f(n) = O(g(n))$

! $f(n) = 10n+5$, $g(n) = \log_{10} n$

$$\left.\begin{array}{l} f(n) = 10n+5 \\ g(n) = \log_{10} n \end{array}\right] \quad 10n+5 \leq 1000 * \log_{10}(n)$$ [Not True for large n values]

$n=10$, $105 \leq 1000*1$

$n=100$, $1005 \leq 1000*2$

$n=1000$, $10005 \leq 1000*3$

$\vdots \quad \vdots \quad \vdots$

$\therefore f(n) \neq O\,g(n)$

$$\boxed{\begin{array}{l} f(n) = O(g(n)) \\ \text{Then, } g(n) \text{ Asymptotically bigger or equal to } f(n) \end{array}}$$

Asymptotic Comparison of functions

(1) DECREMENT FUNCTIONS

$$f(n) = \frac{c}{n},\ \frac{100}{n^2},\ \frac{n}{2^n},\ \frac{\log n}{n}$$

$$\frac{n}{2^n} < \frac{100}{n^2} < \frac{c}{n} < \frac{\log n}{n}$$

(2) CONSTANT FUNCTIONS

$$f(n) = 10 \text{ (const)}$$

(3) LOGARITHMIC FUNCTIONS

$$f(n) = \log n,\ (\log n)^{10},\ \log\log n,\ (\log\log n)^{10},\ \log\log\log n,\ (\log\log\log n)^{10}$$

(4) POLYNOMIAL FUNCTIONS

$$f(n) = n^{0.1},\ n^2,\ \sqrt{n},\ n\log n,\ n^3,\ n^k \text{ (k=const)}$$

$$\{ n^{0.1} < \sqrt{n} < n\log n < n^2 < n^3 < \underset{(k>3)}{n^k} \}$$

(5) Exponential FUNCTIONS

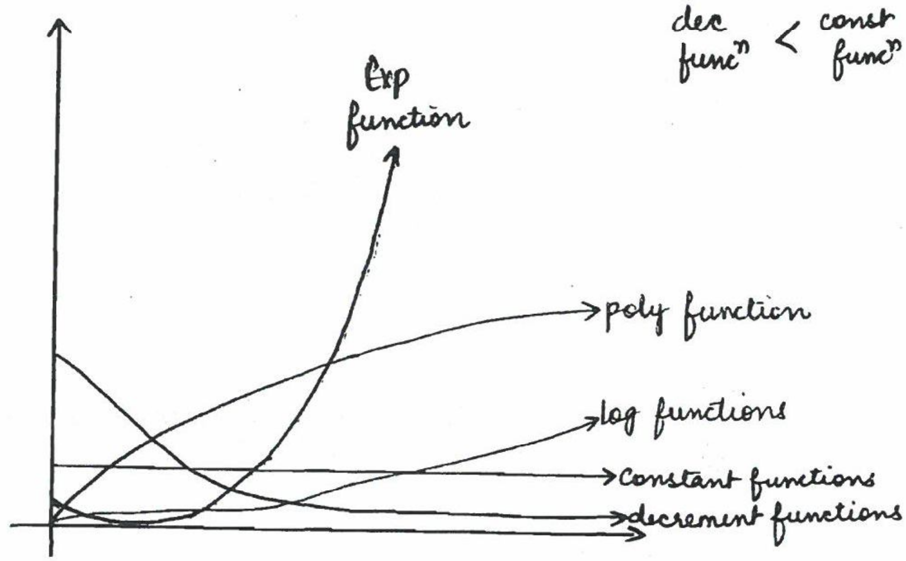
$$f(n) = (1.1)^n,\ (1.01)^n,\ 2^n,\ 3^n,\ n!,\ n^n$$

$$\{ (1.01)^n < (1.1)^n < 2^n < 3^n < n! < n^n \}$$

$$\boxed{a^n = \begin{cases} \text{dec func}, & a<1 \ \& \ a>0 \\ \text{constant}, & a=1 \\ \text{Exp func}^n, & a>1 \end{cases}}$$

$$\boxed{\text{decrement func}^n < \text{Constant func}^n < \text{log func}^n < \text{Poly func}^n < \text{Exp func}^n}$$

dec func^n < const func^n < log func^n < poly func^n < Exp func^n



Write given functions in asymptotic Increasing order:

(a) $\{\underset{E}{(1.5)^n}, \underset{P}{n^2}, \underset{L}{(\log n)^{10}}, \underset{C}{10}, \underset{E}{2^n}, \underset{P}{n^{2.1}}, \underset{P}{n^2 \log n}, \underset{D}{\frac{c}{n}}, \underset{D}{\frac{n}{2^n}}\}$

$$\frac{n}{2^n} < \frac{c}{n} < 10 < (\log n)^{10} < n^2 < n^2 \log n < n^{2.1} < (1.5)^n < 2^n$$

(b) $\log n$, $(\log n)^{10}$, $\log\log n$, $(\log\log n)^{10}$, $\log\log\log n$, $(\log\log\log n)^{10}$, $\log n \cdot \log\log n$, $\log\log n \cdot \log\log\log n$

$$\log\log\log n < (\log\log\log n)^{10} < \log\log n < \log\log n \cdot \log\log\log n < (\log\log n)^{10} < \log n$$
$$< \log\log n \cdot \log n < (\log n)^{10}$$

$(\log\log n)^k < \log n$

$(\log\log\log n)^k < \log\log n$

$\log n < (\log n)^k$

$\log\log n < (\log\log n)^k$

(asymptotically)

LOG PROPERTIES

$\star\ \log_b a = \dfrac{1}{\log_a b}$

$\star\ \log_b a = \dfrac{\log_c a}{\log_c b}$

$\star\ a^{\log_c b} = b^{\log_c a}$

$\star\ \log_b^a n = (\log_b n)^a$

$T(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} \approx \Theta(\log_e n)$

$T(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \cdots + \frac{1}{m} \approx \Theta(\log_e \log_e n)$
(m is largest prime $\leq n$)

$T(n) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} = O(n^2)$

$T(n) = 1^2 + 2^2 + \cdots + n^2 = \Theta(n^3)$

$T(n) = 1^3 + 2^3 + \cdots + n^3 = \Theta(n^4)$

Q Write func^n in asymp. increasing order?

$f(n) = \log\log\log n^k = \log\log[k.\log n] = \log[\log k + \log\log n] \approx \log\log\log n$

$g(n) = \log\log\log^k n = \log(\log(\log n)^k) = \log(k \log\log n) = \log k + \log\log\log n \approx \log\log\log n$

$h(n) = \log\log^k \log n = \log(\log\log n)^k = k.\log\log\log n \approx \log\log\log n$

$s(n) = \log^k \log\log n = (\log\log\log n)^k$

where $k>1$ constant.

$$[f(n) = g(n) = h(n) < s(n)]$$

Q which is false?

(a) $100 \log n = \frac{\log n}{100} = O(\log n)$ True

(b) if $x<y$ then $n^x = O(n^y)$ True

(c) $n^a = O(a^n)$ $a>1$ const True

(d) $\sqrt{\log n} = O(\log\log n) \Rightarrow \sqrt{\log n} = \frac{1}{2}\log n \uparrow \ngtr \log\log n \downarrow \therefore$ False.

Q $f(n) = \begin{cases} \cancel{n^3 \quad 0 \le n \le 1000} \\ n \quad n > 10000 \end{cases}$

$g(n) = \begin{cases} \cancel{\sqrt{n} \quad 0 \le n \le 100} \\ n^2 \quad n > 100 \end{cases}$

for large n values,

$n = O(n^2) \Rightarrow f(n) = O(g(n))$

which is true?

(a) $f(n) = O(g(n))$ and $g(n) = O(f(n))$

(b) $f(n) = O(g(n))$ and $g(n) \neq O(f(n))$

(c) $f(n) \neq O(g(n))$ and $g(n) = O(f(n))$

(d) $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$

Q $f(n) = \log_{10} n$

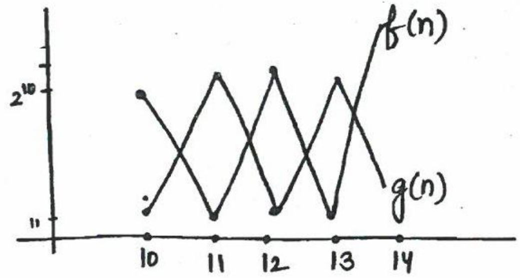
$g(n) = \log_b n^a$

$\Rightarrow g(n) = a \log_b n = a \frac{\log_{10} n}{\log_{10} b}$

$\therefore f(n) = O(g(n))$

& $g(n) = O(f(n))$

Q $f(n) = \begin{cases} 2^n, \text{for even } n \\ n, \text{otherwise} \end{cases}$

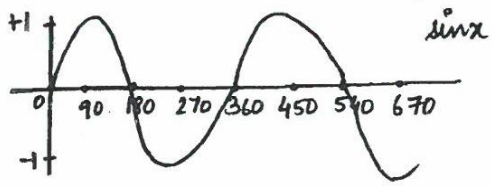


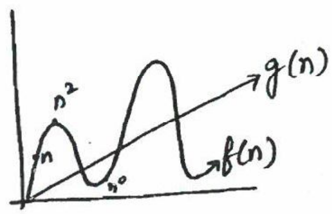
$g(n) = \begin{cases} 2^n, \text{for odd } n \\ n, \text{otherwise} \end{cases}$

they are asymptotically incomparable.

$\therefore f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$

Q $f(n) = n^{1+\sin n}$, $g(n) = \sqrt{n}$





Here also, f(n) and g(n) are not comparable asymptotically.

Thus, $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$

Q $f(n) = n^{1+\sin n}$, $g(n) = n^3$

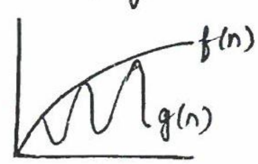
$\leq \sin n \leq 1$

$\therefore f(n) = O(g(n))$

Q $f(n) = n^{2+\sin n}$, $g(n) = n^{1/2}$

Here, $g(n) = O(f(n))$

Q $f(n) = n^2$, $g(n) = n^{1+\sin n}$



$g(n) \leq f(n)$

$\Rightarrow g(n) = O(f(n))$

NOTE:

$f(n)$ $\qquad$ $f(n)+n$ $\qquad$ $f(n)*n$

$f(n) = f(n)+n$ $\qquad$ $f(n) < f(n)*n$

if $f(n) = n^2$ $\quad$ if $f(n) = n^2+n$ [Asymp equal]

$f(n) < f(n)+n$

if $f(n) = \log n$ $\quad$ $\log n < n \log n$ [Asymp greater]