Design & Analysis of Algorithm

(Prof. Parminder Kaur Wadhwa)

Introduction :-

Algorithm and its importance, mathematical foundations :- Growth functions, Complexity Analysis of Algorithms

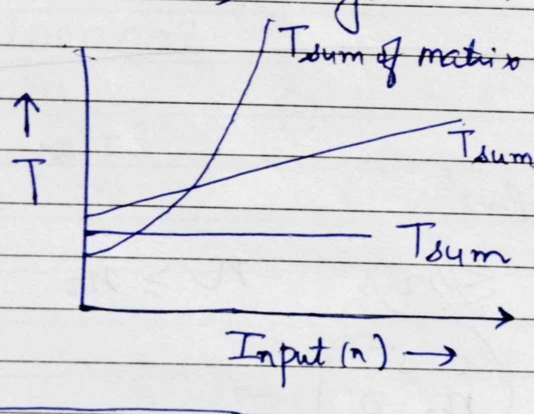
---

Algorithm

Problem P → A1, → A2, → ⋮ ? choose

n → input size

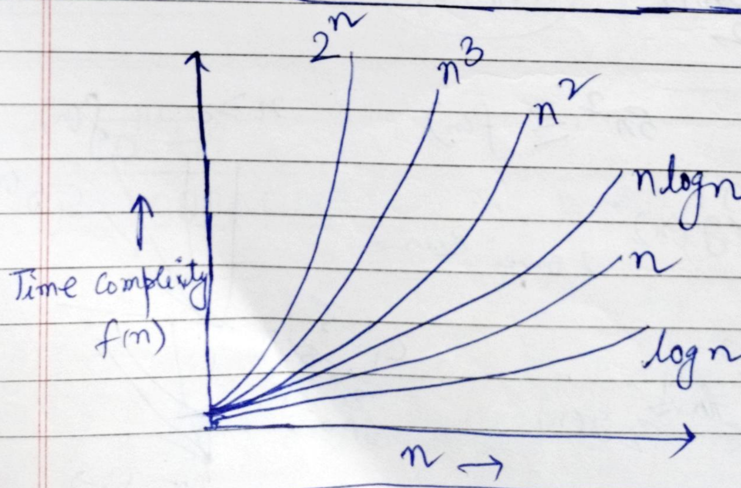
n ↑ , algorithm? behave?



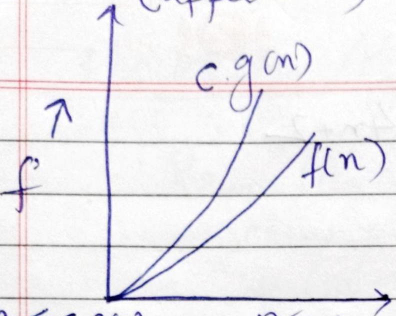
T sum of matrix — for..... for..... $O(n^2)$

T sum of list ⇒ for loop — sum = sum + i — $O(n)$

T sum = a+b — k — $O(1)$

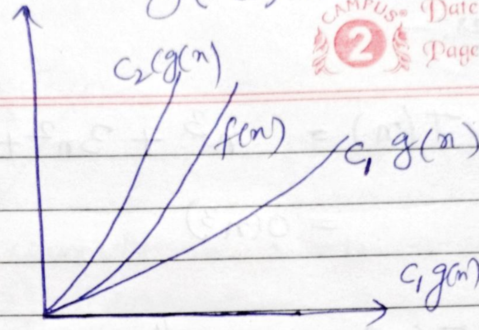


Worst case — Big oh (O) (upper bound)

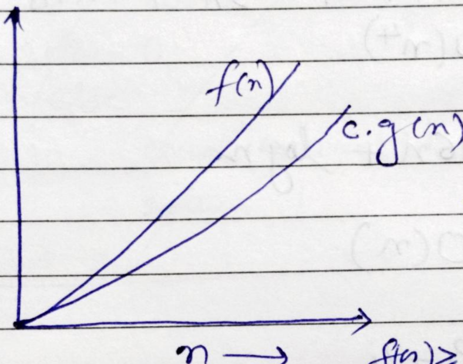


$f(n) \le c.g(n)$

Tight bound (Θ) Theta

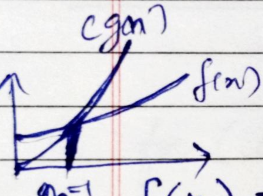


$c_1 g(n) \le f(n) \le c_2 g(n)$



Lower Bound (Ω) Omega

$f(n) \ge c.g(n)$ — Best case



$f(n) = 5n^2 + 2n + 1$ $\qquad 2n \not> 2n^2$

$g(n) = n^2$ $\qquad 1 \not> n^2$

$c = 8 \quad = (5+2+1)$

$f(n) \le 8n^2$ $\quad$ for $(n \ge 1)$

$O(n^2)$

$f(n) = 5n^2 + 2n + 1 = \Omega(n^2)$ $\qquad n \ge 0$

$g(n) = n^2$ $\qquad 2n+1 \ge 1$

$c = 5$ $\qquad 5n^2 \le f(n), \quad n \ge 0$

$n_0 = 0$

$5n^2 \le f(n) \le 8n^2$

$\boxed{c_1 = 5, \; c_2 = 8, \; n_0 = 1}$ $\qquad f(n) = \Theta(n^2)$

Examples:-

1. $T(n) = n^3 + 3n^2 + 4n + 2$

$= O(n^3)$

2. $T(n) = 17n^4 + 3n^3 + 4n + 8$

$= O(n^4)$

3. $T(n) = 16n + \log n$

$= O(n)$

4. $T(n) = 3 + 4$

$\Rightarrow O(1)$

5. $T(n) = a + b$

$= O(1)$

executes | Cost

	times	cost
6. for (i=0; i<=n; i++)	(n+1) times	c_1
{		
// stmts	n times	c_2
}		

$$T(n) = c_1(n+1) + c_2 n$$

$$= c_1 n + c_1 + c_2 n$$

$$= (c_1 + c_2) n + c_1$$

⇒ It is in the form:-

$$T(n) = an + b$$
$$= O(n)$$

i.e. linear relationship

7. **Two for loops** (nested)

```
for (i=0; i<n; i++)
{
  for (j=0; j<n; j++)
  {
    statement }
}...
```

$n+1$

$n(n+1)$

$= n^2$

4

12

9

$n+1 \rightarrow n$

$n(n+1) \Rightarrow \frac{n}{\cancel{n+1}} * n(\cancel{n+1})$

$= n^2$

$n^2+n \Rightarrow$

$n+1 \rightarrow n$

$n^2+n \rightarrow \frac{n}{n+1} * (n^2+n)$

$= \frac{n(n)(\cancel{n+1})}{(\cancel{n+1})}$

$= n^2$

$\Rightarrow O(n^2)$

8. ## Three for loops (nested)

```
for( i=0; i< n; i++)
{
    for( j=0; j <n; j++)
    {
        for( k=0; k <n; k++)
        {
            statement;
        }
    }
}
```

$n+1$

$n(n+1)$

$n^2(n+1)$

n^3

$n+1 \rightarrow n$

$n^3+n^2 \rightarrow \frac{n}{n+1} * (n^3+n^2)$

$\Rightarrow \frac{n(n^2)\cancel{(n+1)}}{\cancel{n+1}}$

$\Longrightarrow O(n^3)$

9. fragments:-

for ($\underline{0 \to n}$)
{

}

$\to (n+1)$ times

$O(n)$

for $(0 \to n)$
{
for ($\underline{0 \to n}$)
{

}
}

$O(n^2)$

$T(n) = O(n) + O(n^2)$

$T(n) \Rightarrow O(n^2)$

Analysis of :—

p. Algorithm to compute Sum

	Cost	Times
$S \leftarrow 0$	c_1	1 time
for $i \Rightarrow 1$ to n do	c_2	$(n+1)$ times
$S \leftarrow S + A[i]$	c_3	n times
return S	c_4	1 time

$$T(n) = c_1 * 1 + c_2 * (n+1) + c_3 * n + c_4 * 1$$

$$= c_1 + \underline{c_2 n} + c_2 + \underline{c_3 n} + c_4$$

$$= (c_2 + c_3)n + (c_1 + c_2 + c_4)$$

$\Rightarrow$ In the form

$$an + b$$

$$= O(n)$$

Date
Page

11. Analysis of Insertion Sort

```
for j ← 2 to length [A]                     c1    n
{   do key ← A[j]                           c2    n-1
       i ← j-1                              c3    n-1
    while (i > 0 and A[i] > key)            c4    Σ_{j=2}^{n} K
    {
       do A[i+1] ← A[i]                     c5    Σ_{j=2}^{n} K-1
          i ← i-1                           c6    Σ_{j=2}^{n} K-1
    }
    A[i+1] ← key }                          c7    n-1
}
```

$$\Rightarrow T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} K$$

$$+ c_5\left(\sum_{j=2}^{n} K-1\right) + c_6\left(\sum_{j=2}^{n} K-1\right) + c_7(n-1)$$

As $\Rightarrow \sum_{j=1}^{n} = \frac{n(n+1)}{2}$ $\quad 1+2+3 \ldots$

$\Rightarrow \sum_{j=2}^{n} \Rightarrow \frac{n(n+1)}{2} - 1$ $\quad 2+3+ \ldots$

$\therefore$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4\left(\frac{n(n+1)}{2} - 1\right)$$

$$+ c_5\left(\left(\frac{n(n+1)}{2} - 1\right) - 1\right) + c_6\left(\left(\frac{n(n+1)}{2} - 1\right) - 1\right) + c_7(n-1)$$

$\Rightarrow an^2 + bn + c \Rightarrow O(n^2)$

Practice Questions :-

1. Do the analysis of selection sort algorithm.

(Hint :- $O(n^2)$)

2. Do the analysis of bubble sort algorithm.

(Hint :- $\rightarrow O(n^2)$)