ALGORITHM

step by step procedure to solve a problem oldest recorded non-trivial algorithm.
Eulid's algorithm for finding GCD of two +ve integers

EULID'S ALGORITHM

1) Input A and B
2) If A & B are equal, either is GCD
3) If A > B, replace A by A - B else B by B - A
4) Go to step 1

Example :-

A	B
45	15
30	15
15	15

15

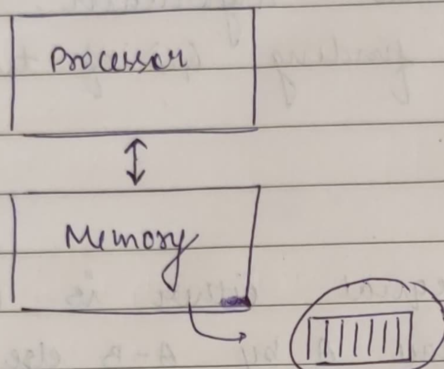
CHARACTERISTICS OF ALGORITHM :-

(1) Finite number of steps
(2) Finite number of inputs
(3) Finite number of outputs
(4) Must terminate in finite time

→ we can have an algorithm with zero inputs ✓
but there must be atleast one output ✓

RAM MODEL (RAM: Random Access Machine here)

This is an abstract model.



Processor is capable of writing as well as reading in the memory.

Memory is made of many cells.

Processor can read a memory cell randomly and also can write in any cell randomly.

OPERATIONS
- Basic operation
- Complex operation.

Basic operation ⇒ addition, substraction, multiplication, division, comparison, memory read/write

Complex operation ⇒ i++ (two operations are being performed i.e. addition & assignment)

PROGRAM

→ specific hardware - processor and memory

→ OS

→ Programming Language

\# Algorithm is judged by its "running time"

→ Each basic operation takes one unit time (assumption)
→ multiplication is repeated addition
→ division is repeated substraction

Example

Q. $i = i+1$ ⇒ In this if we assume basic operation takes 1 unit of time then

addition
assignment } 1 unit + 1 unit = $\boxed{2 \text{ unit}}$ time.

Q. for i = 1 to n   loop
c[i] ← a[i] + b[i]   ⇐ basic operation takes 1 unit time (assumption)
assignment

$\boxed{T(n) = c.n + k}$ → $\underbrace{c[i] \leftarrow a[i] + b[i]}$

i = 1
i++
i < n } steps

addition and assignment for n times (for loop i from 1 to n)

Linear Running Time

TYPES OF ALGORITHM

Deterministic Algorithm

- with respect to time and place it changes
- it changes on different runs
- eg - Linear search

↓

(for a given particular time, the computer will always produce the same output going through the same states)

Non-deterministic Algorithm

- different behaviour on different runs
- eg - Searching (Randomization)

↓

(for same input, the compiler may produce different outputs in different runs)

Example

Q. Concurrent Algorithm (non-deterministic)

→ sum = 0

sum = sum + A[i]

250	250	250	250
sum_1	sum_2	sum_3	sum_4
P_1	P_2	P_3	P_4

↳ $sum = sum_1 + sum_2 + sum_3 + sum_4$

1^{st} run ⇒ P_2, P_3, P_4, P_1

2^{nd} run ⇒ P_1, P_2, P_3, P_4

after step 1 :-

$sum = sum_2 + sum_3$ ...

$sum = sum_1 + sum_2$

↳ not same

QUALITY OF ALGORITHM

Quality of algorithm is decided by two parameters time and space

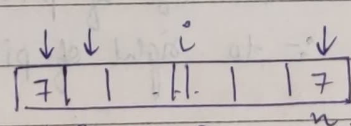
- Time complexity → time taken by an algorithm
- Space complexity → space taken by an algorithm

ANALYSIS OF ALGORITHM

- Best case → Input is provided in such a way that we have minimum time to process
- Worst case → Input is provided in such a way that we have maximum time to process
- Average case → All inputs are equally likely to occur

Example

Linear Search.



key = 7

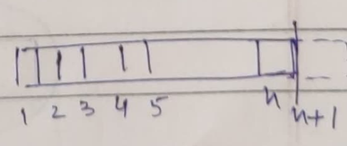
{Searching Problem A[ ], key}

Best case : (constant) $O(1)$ | key with A[i]

Worst case : $O(n)$ (linear)

Average case : $O(n)$ (linear)

$$\left\{ T_n = \frac{1+2+3+ \cdots n}{n} \right\}$$

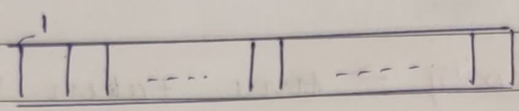


$T_n = \frac{n+1}{2} = \boxed{O(n)}$.

↳ linear function

⇒ NON-RECURSIVE IMPLEMENTATION OF QUICK SORT AND MERGE SORT

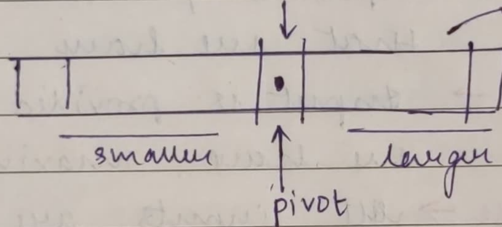
Quick Sort



pivot = A[1] (taking first element as pivot)

now we have to place pivot element in such a way that elements left to it are smaller and elements right to it are larger. This process is called "partition".

partition ⇒

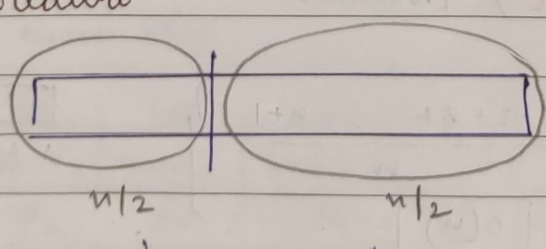


(n) comparisons needed in partition procedure.

after pivot is placed properly, we will repeat this process again and again to two halves :- to left of pivot (among them)
:- to right of pivot (among them)

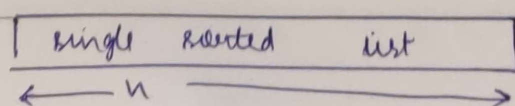
Merge Sort   O(n logn)

Merge Procedure



for every element we are making comparisons

merge after sorting the two halves
↳ O(n)



'O' NOTATION

Consider two functions f & g

large values of N. (N) to R^*

$f \equiv O(g)$ $c > 0$ and $n \geq n_0$

$$\boxed{c\,g(n) \geq f(n)}$$

(1) CONSTANT FUNCTION

$f(n) = 1627$

$c = 1627$ $c\,g(n) = 1627 \times 1$

$g(n) = 1$ $= f(n)$

$\boxed{c\,g(n) \geq f(n)}$ $\boxed{f \equiv O(g)}$

↳ asymptotically for all values of $n \geq n_0$

(2) LINEAR FUNCTION

$f(n) = 3n + 5$

$f(n) \leq 3n + n$; $n \geq 5$

$\leq 4n$ → $g(n)$

↳ c $\{4 \to c \,;\, n \to g(n)\}$.

$\therefore$ $\boxed{n_0 = 5}$, $\boxed{c = 4}$, $\boxed{g(n) = n}$

$$\boxed{f(n) = O(n)}$$

(3) QUADRATIC FUNCTION

$f(n) = 27n^2 + 16n + 25$

$\{c\,g(n) \geq f(n)\}$ $\forall n \geq n_0$

↳ $f(n) \leq 27n^2 + 16n + n$ for $n \geq 25$

$f(n) \leq 27n^2 + 17n.$

$f(n) \le 27n^2 + n \cdot n$ $\{n \geq 17\}$

(But we already had taken $n \geq 25$ in first step and here we took $n \geq 17$, so now we will take the greater value among $n \geq 25$ & $n \geq 17$ i.e. $n \geq 25$)

$f(n) \le 27n^2 + n^2$ $\{n \geq 25\}$

$= 28n^2$

$\boxed{f(n) \le 28n^2}$

$\underbrace{28}_{c}\ \underbrace{n^2}_{g(n)}$

$c = 28$
$g(n) = n^2$
$n_0 = 25$

$\boxed{f(n) = O(n^2)}$

(4) CUBIC FUNCTION

$\boxed{f(n) = O(n^3)}$

(5) EXPONENTIAL FUNCTION

$f(n) = 2^n + 6n^2 + 3n$

$2^n + 6n^2 + 3n \le 2^n + 6n^2 + n \cdot n$ $\forall n \geq 3$

$2^n + 6n^2 + 3n \le 2^n + 7n^2$

{ Also $2^n \geq n^2$ $\forall n \geq 4$ }

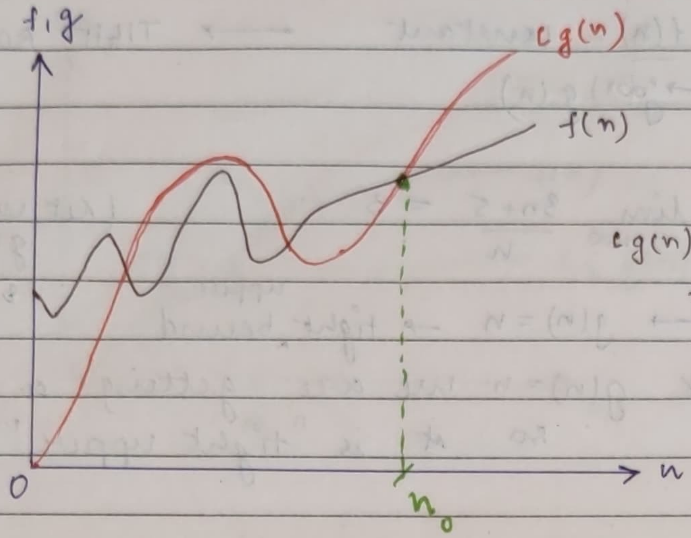
$2^n + 6n^2 + 3n \le 2^n + 7(2^n)$

$= 8 \cdot 2^n$

$\boxed{f(n) \le 8 \cdot 2^n}$ $\forall n \geq 4$

$\underbrace{8}_{c} \cdot \underbrace{2^n}_{g(n)}$

$\boxed{f = O(2^n)}$
$c = 8$
$g(n) = 2^n$
$n_0 = 4.$



$cg(n) \geq f(n)$
$\forall n \geq n_0$.

$cg(n) \geq f(n)$ asymptotically (means n is infinitely large).
as n is infinitely large. (or $n \to \infty$)

$f(n) = 3n+5$
$\leq 3n+n$ $\quad n \geq 5$
$= 4n$
$\underbrace{4}_{c}\underbrace{n}_{g(n)}$.

Now let $c=4$ and $g(n) = n^2, n^3 \dots$
so $cg(n) \geq f(n)$ as order is higher.

{If 4n covers f(n) then $4n^2$ will also cover f(n)}

so it will again asymptotically cover f(n)

Eg. → $f(n) = 3n+5$, $n^2, n^3, n^4, 2^n \dots$

⇒ UPPER BOUND

Least Upper Bound

Tight Upper Bound

$\forall n \geq n_0$

$\underbrace{c\,g(n)}_{\text{upper bound}} \geq f(n)$

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \text{constant}$ ⟶ TIGHT BOUND

Eg :- $\lim_{n \to \infty} \frac{3n+5}{n} = 3$ (Let us take $g(n) = n$)

→ $g(n) = n$ → tight upper bound.

{ For $g(n) = n$ we are getting a constant so it is "tight upper" bound }

:- $\lim_{n \to \infty} \frac{3n+5}{n^2} = 0$ (Let us take $g(n) = n^2$)

↳ loose upper bound.

{ If we have 0 as ans then it is "loose upper bound" }

:- $\lim_{n \to \infty} \frac{3n+5}{n^3} = 0$

↳ loose upper bound.

Tight upper bound for $3n+5 \to n$

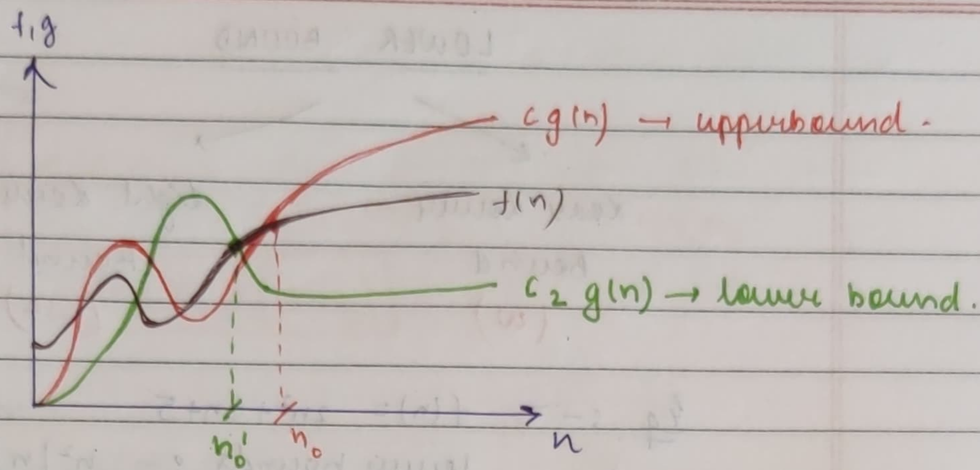
Loose upper bound for $3n+5 \to n^2, n^3$ onward.

Upper Bound. { Loose Upper Bound ⟶ 'o' (small o)
Tight Upper Bound ⟶ 'O' (↳ big O)

$f(n) = 3n+5 = \boxed{O(n)} = o(n^2) = o(n^3) = o(2^n)$

$f(n) = \underbrace{2n^2+3n+5}_{} = O(n^2) = o(n^3)$

↳ $g(n) = n^2 / n^3 / n^k / 2^n$.



LOWER BOUND

$$\boxed{c\,g(n) \le f(n)}, \quad \boxed{\forall n \geq n_0'}$$

(1) CONSTANT FUNCTION

$f(n) = 16$

$c = 15, \quad g(n) = 1$

$g(n) = n^0$

(2) LINEAR FUNCTION

$f(n) = 3n + 5$

$\geq 3n$

c ↲ ↳ g(n) $\quad \forall n$

↳ lower bound.

(3) QUADRATIC FUNCTION

$f(n) = 2n^2 + 3n + 5$

$g(n) = n^2$

$c = 2$

$2n^2 \leq f(n)$

↳ lower bound.

LOWER BOUND

Loose Lower Bound (ω)

Tight Lower Bound. (Ω)

Eg :- $f(n) = 2n^2+3n+5$

lower bounds :- $n^2 \mid n \mid 1$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \text{constant (tight)}$$

$$= \infty \text{ (loose)}$$

$$\lim_{n\to\infty} \frac{2n^2+3n+5}{n^2} = 2 \text{ (tight lower bound)}$$

$$\lim_{n\to\infty} \frac{2n^2+3n+5}{n} = \infty \text{ (loose lower bound)}$$

$$\lim_{n\to\infty} \frac{2n^2+3n+5}{1} = \infty \text{ (loose lower bound)}$$

$\therefore$ Tight Lower Bound $\rightarrow \Omega(n^2)$

Loose Lower Bound $\rightarrow \omega(n), \omega(1)$

'O'	'o'	'Ω'	'ω'
Tight Upper Bound	Loose Upper Bound	Tight Lower Bound	Loose Lower Bound.

'θ' NOTATION

$g(n), c_1, c_2$

$$\boxed{c_1 g(n) \leq f(n) \leq c_2 g(n)} \quad \forall n \geq n_0$$

$$\boxed{f(n) = \theta(g(n))}$$

Eg: $f(n) = 3n+5$

$c=4, \quad g(n)=n$

$\Rightarrow \quad O(n) \qquad \forall n \geq 5$

Now let $c=3, \quad g(n)=n \Rightarrow \Omega(n)$

$\left\{ \lim_{n\to\infty} \frac{f(n)}{g(n)} = \text{constant} \right\}$

$$\boxed{f(n) = 3n+5 = \theta(n) = O(n) = \Omega(n)}$$

$\{ = o(n^2) = \omega(1) \}$

Example

Q. $f(n) = \log(n!)$ give Tight Upper Bound.

→ $f(n) = \log(n!) = \log(n \cdot n-1 \cdot n-2 \cdots 1)$

$= \log n + \log(n-1) + \log(n-2) + \cdots + \log 1$

$\leq \log n + \log n + \log n + \cdots + \log n$

$= n \log n$

$\boxed{f(n) \leq n \log n}$

↳ nlogn is upper bound.

Q. Show that $2^n \neq O(n^m)$

→ Let us assume that $2^n = O(n^m)$

So, $c\,g(n) \geq f(n)$

$c \cdot n^m \geq 2^n$

$c \geq \frac{2^n}{n^m}$

constant

↳ varying depending on n

So contradiction

$\therefore\ 2^n \neq O(n^m)$

Q. $f(n) = a_m n^m + a_{m-1} n^{m-1} + a_{m-2} n^{m-2} + \dots + a_1 n + a_0$

and coefficient $a_m > 0$.

$$\left.\begin{aligned} &= O(n^m) \\ &= \Omega(n^m) \\ &= \theta(n^m) \end{aligned}\right\}$$