Algorithmic Analysis - 1

17 June 2022 22:25

Q Algorithm:-
An algorithm is step by step instruction to solve a given problem.

E.g. Prepare biriyani

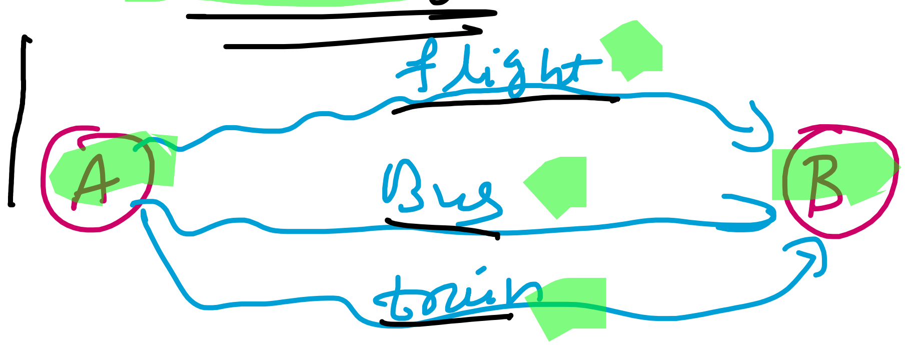
Algorithm:-
1. Get a frying pan
2. Take the oil, rice, veg
3. Turn on the stove
4. Cook it for 30 min.

Problem:- Reverse the arr.
arr - 1, 2, 3, 4, 5
O/P - 5, 4, 3, 2, 1

Algo:-
1. Swap 1st & last elem
2. Swap 2nd & 2nd last elem.

1. Read the prob
2. Devise algo
3. Code algo

10% coding
90% algo



Speed	Cost	Comfort
→ flight	Bus	
→ train	train	
→ Bus	flight	

Q Goal of algorithm analysis
→ The goal of Algo anal. in term of
→ Time
→ Memory / Space

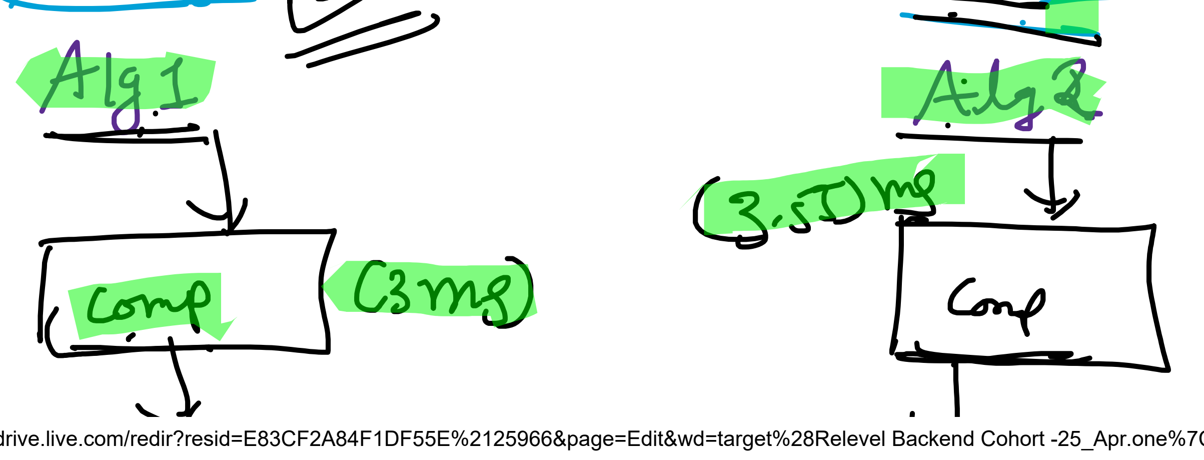
Time ↑↑
Space ↑↑

Coding Round
1. correct sol^n
2. Timing Complexity
3. Space complexity

Q How to go algo. anal.

I. Experimental analysis

Aradhya   Joseph



o/p o/p

Disadvantages:- Aradhya → 16gb ram
Joseph → 4gb ram

Sol^n → machine dependent

2. Asymptotic analysis

i ⇒ not rely on machine

→ compare alogrithms based on the input rate of growth.

Reverse an array, n = length

Aradhya Sol^n Algo1	Joseph Sol^n Algo2
n=5 3ms	3ms n=5
n=100 3.1ms (comp steps)	3.1ms n=100
n=10000 3.5ms	n=10000 10min

① Iterate an array : for(i=0; i<n; i++) i<5
② Compare two values if(x<y) ⇒ ①

n=5

1 ⇒ 5 comp. steps → comp. step.
2 ⇒ 1 step

③ Types of measurement {Alg1}

① worst case }
② Average case }
③ Best case } ⇒

A — flight → B

1. Best case - always on time
2. Avg - avg time taken A→B
3. worst - extreme delay

worst case

Ex. [x = 5 + (15*20)].

n=5, n=100

Constant
1 ✓ ① 15 * 20
2 ✓ ② 5 + 300
3 ✓ ③ x = 305

300 }} 100 }} 10min

Constant time complexity O(3) ≃ O(1)

it represent big-oh notation
↓
worst cases

Q
1 $x = 5 + (15 * 20)$ // $O(3)$
2 $y = 15 - 2$ // $O(2)$
3 Print $x + y$ // $O(2)$

$O(3) + O(2) + O(2)$

$\approx O(1)$ → worst case

Q
for(i=0; i<n; i++) $O(n)$
{
print(i) $O(1)$
}
$O(n+1) = \boxed{O(n)}$

n=5 5 steps
0
1
2
3
4

n=8 8 steps
0
1
2
3
4
5
6
7

$\boxed{n+1}$

n=1 Best case
$O(1)$

$O(n+1)$ $O(n-1)$ $O(n+1)$

$O(n)$

$O(n/2)$ $O(n + \frac{1}{2})$

$O(n)$

for(i=0; i<n; i++) // 5 times n=5 $O(5)$
{
{print(i) // 1 time (1) $O(5) * O(1)$
}
$O(n)$

$\boxed{n=1}$
$O(1)$

0 → 0
1 → 1
2 → 2
3 → 3
4 → 4

Print(i)} $O(1)$
n=5

1m
n = 10000

$O(2+n)$ $O(\underline{20000})$

??? O → worst case

$\cancel{2}n \approx n$ $\boxed{\approx}$

1. Ignore Constant

$O(5n) \approx O(n)$

Why? Because when n becomes large, 5 does not matter

n $\cancel{5}n$

n = 10,00,000 } (3.12 ms)
5n = 50,00,000 } (3.19 ms)

$\approx n$

$5n$ $\cancel{5}(n) \approx O(n)$
$\frac{1}{2}n$ $\times \frac{1}{2}(n) \approx O(n)$

```
for(i=0; i<n; i++)
{
    print(i)
}
```
$O(n)$

```
for(j=0; j<n/2; j++)
{
    print(j)
}
```
$O(n/2)$
$O(\cancel{\tfrac{1}{2}} * n)$
$O(n)$

$\approx \boxed{O(n)+O(n)}$ $\Rightarrow \boxed{\cancel{2}\,O(n)}$
$\approx \boxed{O(n)}$

Q
y = 5 + (15*20) → $O(1)$

```
for(i=0; i<n; i++)
{
    print(i)
}
```
$O(n)$

$\cancel{O(1)} + O(n) \Rightarrow O(n)$

$O(1) < O(\log n) < O(n) <$
$O(n \log n) < \boxed{O(n^2)} < O(2^n)$
$< O(n!)$

Q for(i=0; i<n; i++) // O(n)

```
{
  for(j=0 j<i j;j++) {
      Print(i*j)
  }                      } O(n)
}
```

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

Q

```
if (x<0)
{
   // O(1)
}
else if (x>0)
{
   // O(logn)
}
else {
   // O(n^2)
}
```

x=-1 → O(1)

x=5 → O(logn)

x=0 → O(n^2)

Worst

$\boxed{\text{MRF}}$

O(n^2)

O(1)

O(logn)

O(n) O(n/2)

O(1) O(n/2) => O(n)

Q

```
let a = n;
  while(a>0){
     a = ParseInt(a/2)
  }
```

$\boxed{TC \rightarrow}$

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

   }
}
```

O(n^2)

i n

n

n+i

{1+2+3+4+ ---- +n}

(n^2+n)/2

0+1+2+3+ --- + n

```
let n = 5;
for(i=0;i<n;i++){ //O(n)
  for(j=0;j<i;j++){
    console.log(i,j)
```

```
    }
}
```

Q

```
let a = n;
while(a > 0){
  a = ParseInt(a/3)
}
```

$\boxed{TC \rightarrow}$

$\boxed{n=16}$
a=16
a=8
a=4
a=2
a=1
a=0

n=8
a=8
a=4
a=2
a=1
a=0

n=20
a=20
a=10
a=5
a=3
a=1
a=0

$\log_3 n$

$\log_2 16 \Rightarrow 4 + 1 = 5$

$\log_2 n + 1 \approx$

$\boxed{\log_3 n}$ $\log_2 n$

$\log_2 16 + 1$
↓
4

$\log_2 16 + 1 \Rightarrow \log_2 16$

Q

```
for(i=0; i< n; i++)   // O(n)
{
  if ( ) {
    if ( ) {
      // O logn
    }
    O(n)
  }
  else {
    O(n)
  }
}
```

O(logn)

O(n)

O(n)

O(n)

O(n) * O(n)

$\boxed{\approx O(n^2)}$

```
for (          )   → O(n)
  if ( ) {
    //O(n)
```

O(nlogn)

}

else {

// O(nlogn)

}

}

O(nlogn)

O(n) * O(nlogn)

$\boxed{O(n^3 \log n)}$

Reliance

Future Retail

MRP → MRP

MRP

* Notations :

1. Big-O notation — O → worst cases.
2. Theta notation — Θ(n) Avg cases
3. Omega notation — Ω( ) → Best cases

1. Big-oh notation

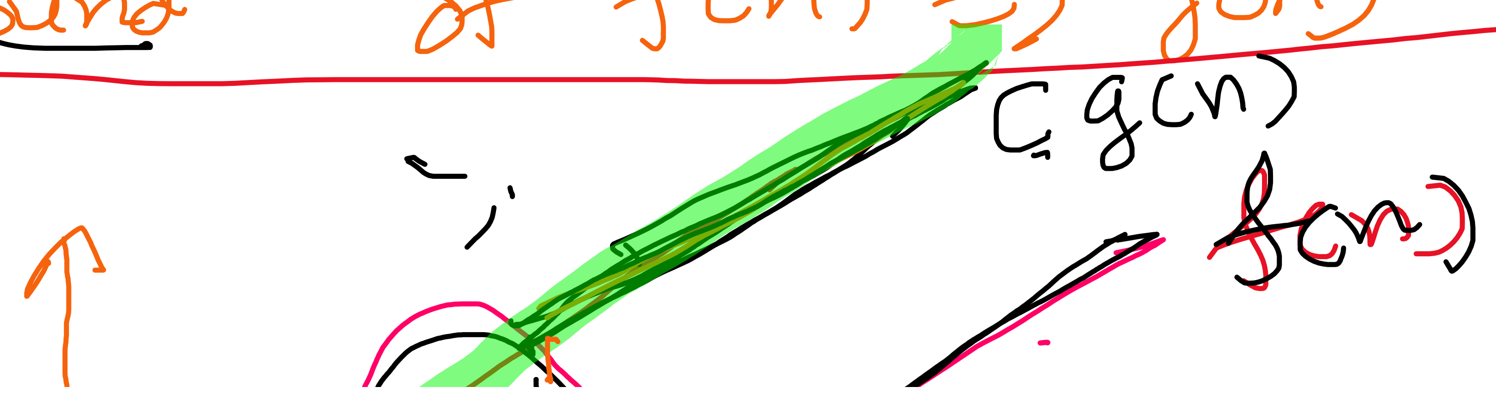
→ gives you tight upper bound of algo.

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

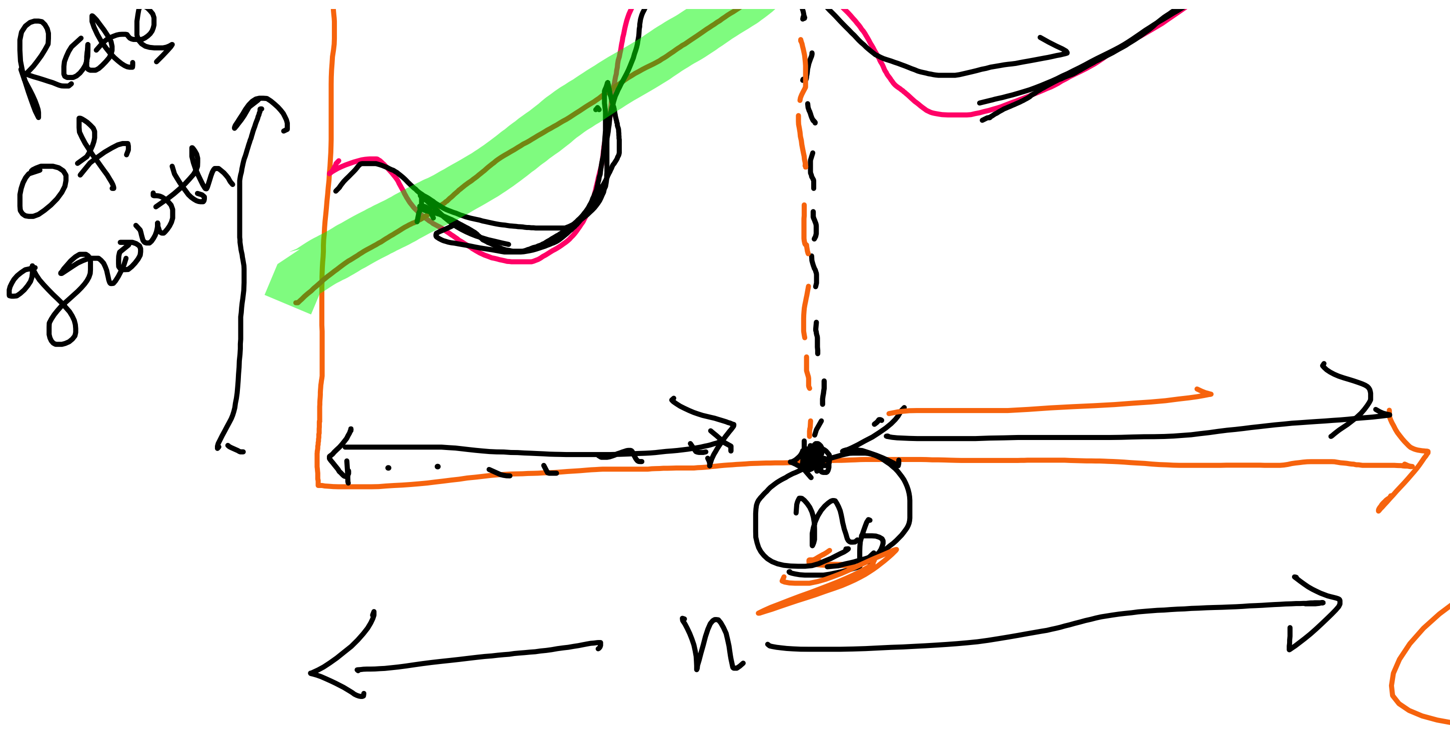
at larger value of n, the upper bound of f(n) ⇒ g(n)

MRP → 10 Lakh

MRP → 25 Lakh



Worst Case



$n=1$

$n \geq 100$

$1 + 100 + 10 + 50$

168

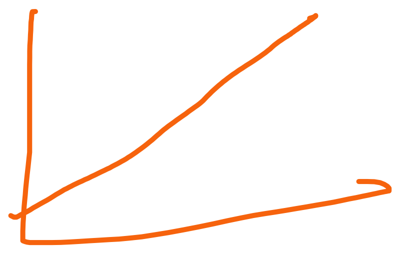
$f(n) = n^4 + 100n^3 + 10n + 50$

$n^4 \rightarrow g(n)$

$O(g(n))$ { $f(n)$: there exist positive constants n_0, c such that

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

for all values $n \geq n_0$ }



$f(n) = 3n+2$

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

for all values $n \geq 2$ $\quad n \geq n_0$

$n=3 \Rightarrow 3*2+2 \leq 4*3$

$8 \leq 12$

$n=1 \quad 3*1+2 \leq 4$

$6 \leq 4$ ✗

$4n$

$3n+2 = O(n)$

$\boxed{3n+2 \leq 4n}$

$c = 4$

$n=1$

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

$5 \leq 4$ ✗

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

$\boxed{n = g(n) \quad c = 4}$

$n = 2 \quad 8 \le 8$

$n = 3$

$9+1 \le 12$

$10 \le 8$

Q. $f(n) = n^2 + 1$

$g(n) =$

c

n_0

$n^2 + 1 \le$ ___

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

$\boxed{n^2 + 1 \le 2n^2}$

$C = 1$

$n_0 = 1$

$O(n^2)$ where all values ≥ 1

$n = 1 \quad 1+1 \le 1$

$2 \le 1$ ✗

$n = 2 \quad 4+1 \le 4$

$5 \le 4$

$g(n) = O(n^2)$

$C = 2$

$n_0 = 1$

$1+1 \le 2 \quad n=1$

$2 \le 2$

Q. $\underbrace{n^2+1}_{f(n)} \le 1 n^2 \quad n=1$

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

$\boxed{n^2 + 1 \le 2 n^2} \quad n$

$C = 2$

$g(n) = n^2$

$n_0 = 1$

C

n_0

$\boxed{g(n)}$

$n=2 \quad 4+1 \le 2 * 4$

$\boxed{5 \le 8}$

$n=1 \quad 1+1 \le 2$

$n=1$ $\quad 1+1 \le 1$

$2 \le 2$

$n=3$ $\quad 4+1 \le 4$

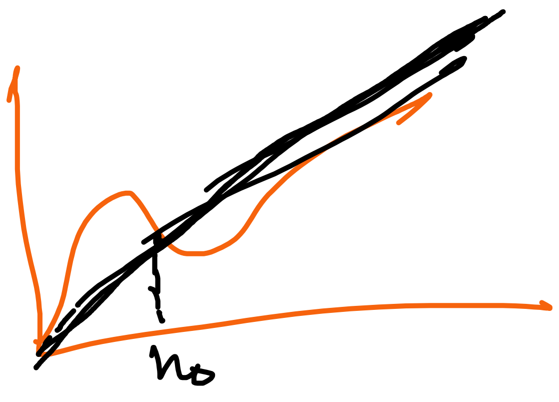
$\boxed{5 \le 5}$ X

$n=3$ $\quad 9+1 \le 18$

$10 \le 18$

$n=3$

$\boxed{n \ge 2}$



Q

$100n + 5$

$g(n) = n$

$c = 105$ all $n \ge 1$

$n_0 = 1$

$\boxed{50n^2}$

$cg(n)$

$3(50n)$

$150n$

$\boxed{100n + 5 \le 105n}$

$n=1 \quad 105 \le 105$

$n=2 \quad 205 \le 210$

$n=3 \quad 305 \le 315$