10-09 Revision Class

09 September 2022 23:06

Algorithm - Step by Step Process to solve your problem.

A B A T B

Space ⇒ Space Complexity

Time ⇒ time comp.

1. Read Prob
2. Choose algo./logic/process. 80%
3. Implement algo. 10%

Practice.

1. Leetcode
2. Hackerrank
3. Hackerearth
4. InterviewBit
5. GFG

Easy Medium Hard

topics forum

Array

List

Excel

Topics

Practice Leetcode

Algo 25th

1 2 3 4 5 ... 10 A

11 ... 20 B

1 ... 30

80%

Doubts

TAs

Doubt class

Recurrence rel^n

tougher topics

time

recursive sol^n exercise

Mon 2 W S

Sorting

Sorting

time Complexity

① Experimental anal.

Akshay → ← Sudhakar

mac Pen-Paper

A B

3sec.

Compatible xxx

Systems should be same

∝ machine dependend.

F1 racer ⇒ 1st bis

Loop

1M+

2. Asymptotic analysis. ∝ no. of Computer Steps

∝ machine ×

Pen-Paper

O( ) n

Cloud → Googledrive
Onedrive
AWS
Azure

flipkart.com } Address
domain virtual address
Aneesh → Address

Hosting

flipkart

eafiles 1 gb

100

100

10 × 10

1 gb

Physical server

Hosting
1 gb x
HD SSD horizontal
10 gb 100 gb

1 gb → 10 gb → 100 gb Scale

Scalable

Cloud

Virtual space
1 gb

Virtual machine 1 gb → 10 gb → 100 gb Vertical scaling
easy scaling
Os
i7
Scalable
IT
5000+
Networking
mac
iphone → mac
RAM
Space
machine
10
click

Sudhakar
Alg1
no. of Cs

Sudhanshu
Alg2
no. of Cs

n = 5 } 5
n = 10 } 50
n = 100 } 1000

Analyze
max

1. Best (Ω)
2. Avg (Θ)
3. Worst (O)

I/P 010100111 } ⇒ Best ⇒ (1)
O/P 000011111
000011111 } O

Worst
{ I/P.

111111 0000

Avg.

11000110

Rules:-

$y = 1+(2+3)$ } n

n=100 n=3 n=1000

3 steps

O(1)

1. Ignore const.
2. Statements => Sequential } + x y → x+y
3. Nested x[ ]y => x·y
4. Ignorable. $n^2 + n$

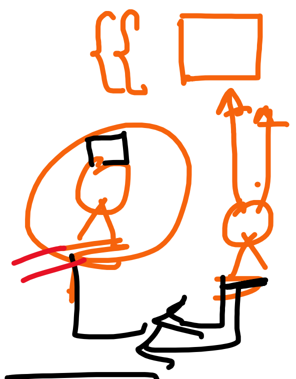
longest

n = 3 9 3×2×1 = 6

n = 10 1000+ $\frac{10\times9\times8\times7\times6\times5\times4}{3\times2\times1}$

	0	1	2	3	4
0					16
1	32	64	128	256	512
2	1024	-	-	-	-
3		-	-		-
4					

Combinations



1 Twice

1 Piece

30 th

31 th

tracks

10/15

```
a=0   b=0
for(i=0; i<n; i++)
{
    a=a+rand();
}
for(j=0; j<m; j++)
{
    b=b+rand();
}
```

1. TC => O(N×m) SC => O(1)
2. TC => O(m) SC => O(1)
3. TC => O(N+m) SC => O(N+m)
4. TC => O(N+m) SC => O(1)

no. of f^n calls

→ call stack

Space Comp

```
for(i=0; i<n; i++)        n
{
    for(j=n; j>1; j++)
    {
        a = a+i+j;        n
    }
```

TC → n^3

SC → 1

}   n:

```
let i,j,k=0;
for(i= n/2 ; i<=n; i++)
{
    for (j=2;j<=n ;j =j*2) {
        k= k+n/2;
    }
}
```

n/2 (for) · logn (for)

n/2 * logn

$n \times \boxed{\frac{1}{2}} \times \log n$

e8 &&?

$\boxed{n\log n}$

$\boxed{TC \\ SC}$

↓ ↑ Generalize

$SC \Rightarrow 1$

$\boxed{\log 16} \Rightarrow 4$

n=16

n=32

$i=8 \Rightarrow j \Rightarrow 2, 4, 8, 16$ } logn

$j = 2, 4, 8, 16, 32$

$\log 32 \Rightarrow 5$

$\log 36 \Rightarrow 6$

```
for(i= n/2 ; i<=n; i++)
{
    for (j=2;j<=n ;j =j/ 2) {
        k= k+n/2;
    }
}
```

∞

$n=8 \Rightarrow j = 2, 1, 1/2, 1/4, 1/8, 1/16$

```
a=0; i=n
while (i>0) {
    a= a+i;
    i =(i/2);
}
```

a. O(n)

b. O(sqrt(n))

c. O(n/2)

d. O(logn)

Pangesnt

> 0

$\boxed{0}$. - - - - - > 0

$n = 16 \Rightarrow i= 16/8/4/2/1/$

0.5

1/4 1/8 1/16

1. Substitution method
2. Master th^m
3. Recursion tree

}

Q

$$T(n) = \begin{cases} 1, & \text{if } n=1 \\ n+T(n-1) & \text{if } n>1 \end{cases}$$

$\boxed{T(1)=1}$

$T(n) = n+T(n-1)$

$T(n) = n * T(n-1) - (1)$ ←

$T(n-1) = (n-1) + T(n-1-1)$

$T(n-1) = (n-1) * T(n-1-1)$

$T(n-1) = (n-1) * T(n-2)$ — ②

$T(n-2) = (n-2) * T(n-2-1)$

$T(n-2) = (n-2) * T(n-3)$ — ③

→ Put ② => ①

$T(n) = n * (n-1) * T(n-2)$

$T(n) = n * (n-1) T(n-2)$ — ④

→ Put ③ => ④

$T(n) = n * (n-1) * (n-2) * T(n-3)$

$= n * (n-1) * (n-2) * \ldots$ $T(n-(n-1))$

$n \times (n-1) \times (n-2) \ldots$ $T(1)$

$n \times (n-1) \times (n-2) \times (n-3) \times 4 \times 3 \times 2 \times 1$

$n \times n\left(1-\frac{1}{n}\right) \times n\left(1-\frac{2}{n}\right) \times n\left(1-\frac{3}{n}\right)$

$\ldots \times n\left(\frac{4}{n}\right) \times n\left(\frac{3}{n}\right)$

$\times n\left(\frac{2}{n}\right) \times n\left(\frac{1}{n}\right)$

$2 \times 2 \times 2 \times 2$

2^4

$n^n \Rightarrow n!$

Q. $T(n) = \begin{cases} 1 & ; n = 1 \\ 2T\left(\frac{n}{2}\right) + n & , \text{otherwise} \end{cases}$ $T(1) = 1$

$T(n) = 2T\left(\frac{n}{2}\right) + n$ — ①

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

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

$2 \times \frac{n}{2}$

$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \frac{n}{4}$ — ③

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

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

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

$$= 2^2 \left[2T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 2n$$

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

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

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

$$= 2^5 T\left(\frac{n}{2^5}\right) + 5n$$

$$2^k T\left(\frac{n}{2^k}\right) + kn$$

$$T(1) = 1$$

$$\boxed{\frac{n}{2^k} = 1}$$

$$\boxed{n = 2^k}$$

$$\log_a a^b \Rightarrow b \log a$$

$$\log_a a \Rightarrow 1$$

$$\log n = \log 2^k$$

$$\log n = k \log_2 2$$

$$\log n = k$$

$$2^k \, T(1) + kn$$

$2^k + kn$

$n + kn$

$n + \log n \times n$

$n + n\log n$

$\approx O(n\log n)$

$T(n) = \begin{cases} 1, & n = 1 \\ T(n-1) + \log n, & n > 1 \end{cases}$

$T(n) = T(n-1) + \log n$ — (1)

$T(n-1) = T(n-2) + \log(n-1)$ — (2)

$T(n-2) = T(n-3) + \log(n-2)$ — (3)

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

$T(n) = T(n-3) + \log(n-2) + \log(n-1) + \log n$

$= T(n-4) + \log(n-3) + \log(n-2) + \log(n-1) + \log n$

$= T(n-k) + \log(n-(k-1)) + \log(n-(k-2)) + \log(n-(k-1)) + \log n$

$T(1) = 1$

$n - k = 1$

$k = n-1$

$= T(n-(n-1)) + \log(n-(n-1-1)) + \log(n-(n-1-2)) + \log(n-(n-1-3)) + \log n$

$= T(1) + \log 2 + \log 3 + \log 4 + \log n$

$= 1 + \log 2 + \log 3 + \log 4 + \dots + \log n$

$= 1 + \log(2 \times 3 \times 4 \times 5 \times \dots n)$ 50 50×1

$= 1 + \log(1 \times 2 \times 3 \times 4 \times 5 \times \dots n)$

$= 1 + \log(n!)$

$= 1 + \log(n!)$

$n \times (n-1) \times (n-2) \times \cdots \times 1$

$n \times n \times n \times n \times \cdots$

n^n

$= 1 + \log(n^n)$

$= 1 + n \log n$

$\boxed{\simeq O(n \log n)}$