

Ayush Sharma \Rightarrow 2019 CSE, IIT Patna

Strand $\downarrow\downarrow$ life Sciences

Scaler Academy
(SDE + Instructor + Module head)

↓

↓

Language → Intermediate → DSA(1, 2, 3, 4)

(Beginners)

↓

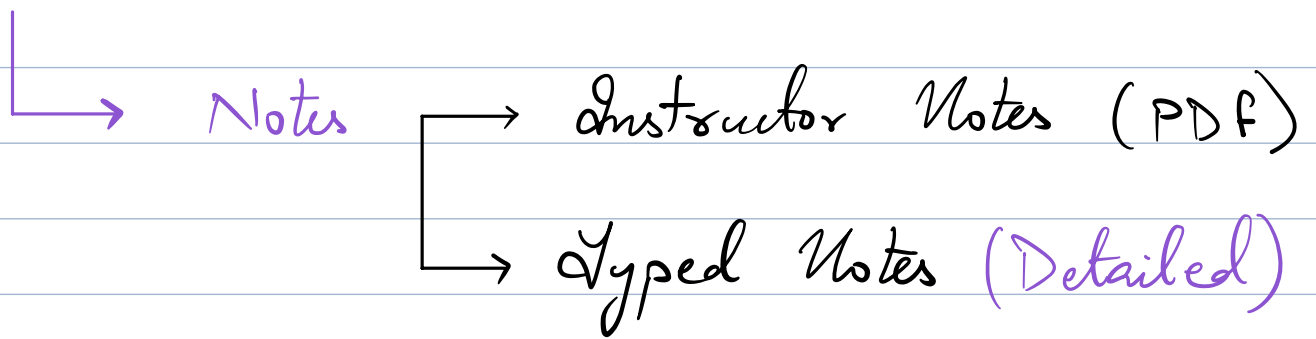
Introduction
↳
Problem
Solving
(1 month)

DSA 4.2

Live lectures \Rightarrow Mon, Wed, Fri \Rightarrow 9PM
Tue Thr Sat/Sun \Rightarrow PSP

Class starts at 9:05 PM

→ Recordings will be present on Dashboard. (6 months after the course ends)



live lecture (Pseudo Code)

Assignments
⇓

Additional Problems

Problem Solving Percentage $\Rightarrow \geq 85\%$
(PSP)

⇓
$$\frac{\text{Total no. of Assignment Ques. Solved}}{\text{" " " " " Unlocked.}}$$

Intermediate Module

How to solve a DSA problem

Build problem solving approach

- 1) Intro. To Problem Solving
- 2) Time Complexity
- 3) Intro. to Arrays
- 4) Prefix Sum
- 5) Carry forward & Subarray
- 6) Sliding Window & Contribution Technique
- 7) Memory Managements
- 8) 2D Matrices
- 9) Bit Manipulation Basics
- 10) Strings
- 11) Interview Problems.

Contest \Rightarrow 1.5 hrs , 4 Questions
9 to 10:30 PM
Pass \Rightarrow 3/4

\Rightarrow Within the class timings

10:30 to 11:30 PM \Rightarrow Contest Discussion

Agenda

Power of making observations.

- 1) Count the factors
 - 2) Optimise
 - 3) Prime Number
 - 4) Sum of first N natural no's
 - 5) Iterations
 - 6) How to compare two algorithms.
-

Q What is a factor ??

⇒ If i is a factor of N , i divides N completely & leaves a remainder 0.

Q How to check programmatically

$\%$ → Modulo operator

⇒ if $(N \% i == 0)$

Eg $12 \% 3 = 0$

⇒ 3 is a factor of 12.

Q Given a no. $N > 0$. Find the count of factors of N .

Q $N = 24 \Rightarrow 1, 2, 3, 4, 6, 8, 12, 24$

factors of $N \Rightarrow [1, N]$

Ans = 8

Q $N = 10 \Rightarrow 1, 2, 5, 10$

Ans = 4

Solⁿ 1) Brute force Solution

\Rightarrow Try all possibilities

Min factor of $N = 1$

Max. factor of $N = N$

⇒ for every no. from 1 to N , we check if it is a factor of N .

Pseudo Code

function countFactors (N) {

 count = 0;

 for ($i=1$; $i \leq N$; $i++$) {

 if ($N \% i == 0$) {

 count = count + 1;

 }

 }

 return count;

}

} N
iterations

⇒ Your code runs on servers.

1 GHz processing power.



Capacity of running $\sim 10^8$ iterations in 1 sec.

N	iterations	Exec. Time
10^8	10^8	1 sec
10^9	10^9 $= 10 \times 10^8$	10 sec
10^{18}	10^{18} $= 10^{10} \times 10^8 \xrightarrow{1 \text{ sec}}$	10^{10} sec 10000000000 sec ≈ 317 years

2) Optimise

$$N = 24$$

$$\begin{array}{r} 4 \\ 6 \overline{) 24} \\ \underline{24} \\ 0 \end{array} \Rightarrow 24 = 6 \times \textcircled{4} \quad \downarrow \quad \frac{24}{6}$$

i	N/i
1	24
2	12
3	8
4	6
6	4
8	3
12	2

Factors are repeating

Eg $N = 100$

i	N/i	
1	100	} $i \leq \frac{N}{i} \Rightarrow$ New factors
2	50	
4	25	
5	20	
10	10	
20	5	
25	4	
50	2	
100	1	

$$i \leq N/i$$

\Rightarrow Multiply i both sides

$$i \times i \leq N$$

$$i^2 \leq N$$

\Rightarrow Take $\sqrt{}$ both sides

$$i \leq \sqrt{N}$$

Factors of $N \Rightarrow [1, \sqrt{N}]$

$N = 100 \Rightarrow [1, 10]$

i		count
1	$\Rightarrow 100 \% 1 == 0$	$+ 2 (1, 100)$
2	$\Rightarrow 100 \% 2 == 0$	$+ 2 (2, 50)$
3	$\Rightarrow 100 \% 3 \neq 0$	\times
4	$\Rightarrow 100 \% 4 == 0$	$+ 2 (4, 25)$
5	$\Rightarrow 100 \% 5 == 0$	$+ 2 (5, 20)$
6	$\Rightarrow 100 \% 6 \neq 0$	\times
7	$\Rightarrow 100 \% 7 \neq 0$	\times
8	$\Rightarrow 100 \% 8 \neq 0$	\times
9	$\Rightarrow 100 \% 9 \neq 0$	\times
10	$\Rightarrow 100 \% 10 == 0$	$+ 2 (10, 10)$ $+ 1 (10)$ if $(i == N/i)$

Pseudo Code

function
 $i \leq \sqrt{N}$

function countFactors (N) {

count = 0;

for (i = 1; $i \times i \leq N$; i++) {

if (N % i == 0) {

if (i == N/i) {

count = count + 1;

} else {

count = count + 2;

}

}

}

return count;

}

\sqrt{N} iter.

N

iterations (\sqrt{N})

Exec. Time

10^{18}

$\sqrt{10^{18}}$

$= 10^9$

$= 10 \times 10^8$ iter.

10 sec.

Q Given a no. N . Check if it is prime ??

→ Count of factors = 2

Prime No \Rightarrow Divisible by only 1 & itself

7 \Rightarrow 1, 7

5 \Rightarrow 1, 5

11 \Rightarrow 1, 11

Basic Maths Properties

Range $[a, b]$ (a, b)

$[,] \Rightarrow$ Including

$() \Rightarrow$ Excluding

$[a, b)$

Q

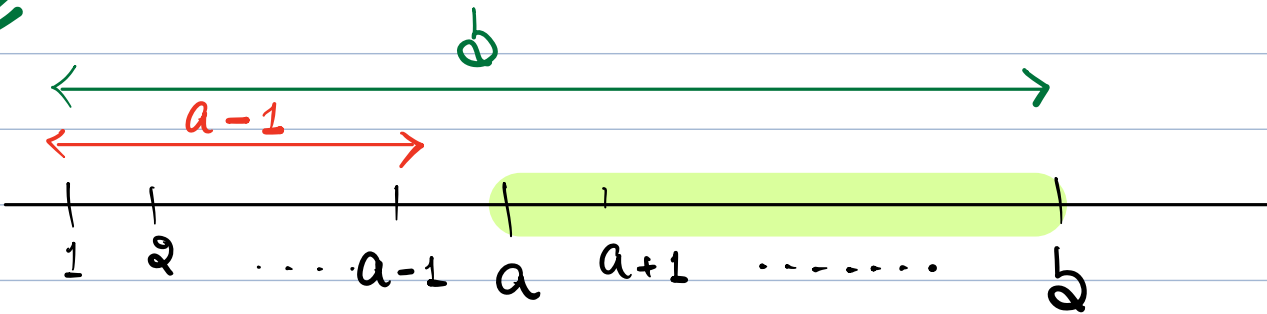
Range $[3, 10]$

3, 4, 5, 6, 7, 8, 9, 10

8

Q

Range $[a, b]$



$$b - (a-1) = \underline{\underline{b-a+1}}$$

Ans = $b - a + 1$

~~Ans~~

Q

$$S = 1 + 2 + 3 + 4 + \dots + 100 = ??$$

Sum of 1st
N natural no. = $\frac{(N)(N+1)}{2}$

~~are~~

Iterations

⇒ The no. of times a loop runs.

1
for (i = 1 to N) &
if (i == N) & break &
}

$$i \Rightarrow [1, N] = N - 1 + 1 = \underline{N}$$

2
for (i = 0 to 100) &
// // //
}

$$i \Rightarrow [0, 100] \Rightarrow 100 - 0 + 1 = \underline{\underline{101}} \text{ Ans}$$

func()

for (i = 1 to N)

//

{

for (i = 1 to M)

//

{

}

} i \Rightarrow [1, N] = N

} i \Rightarrow [1, M] = M

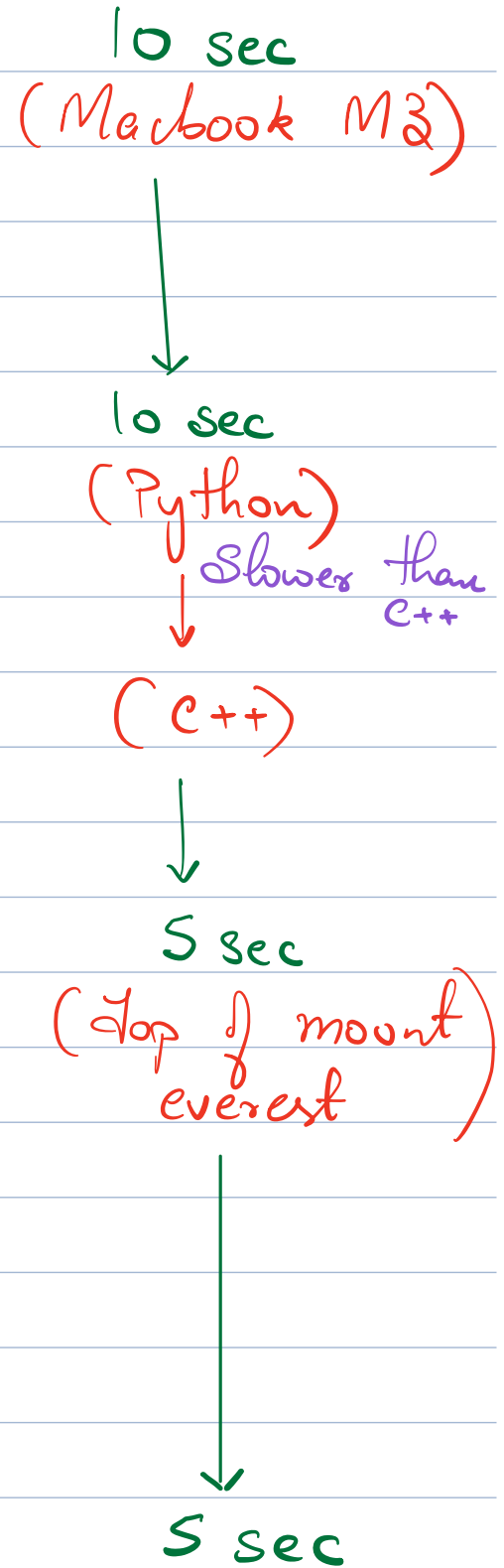
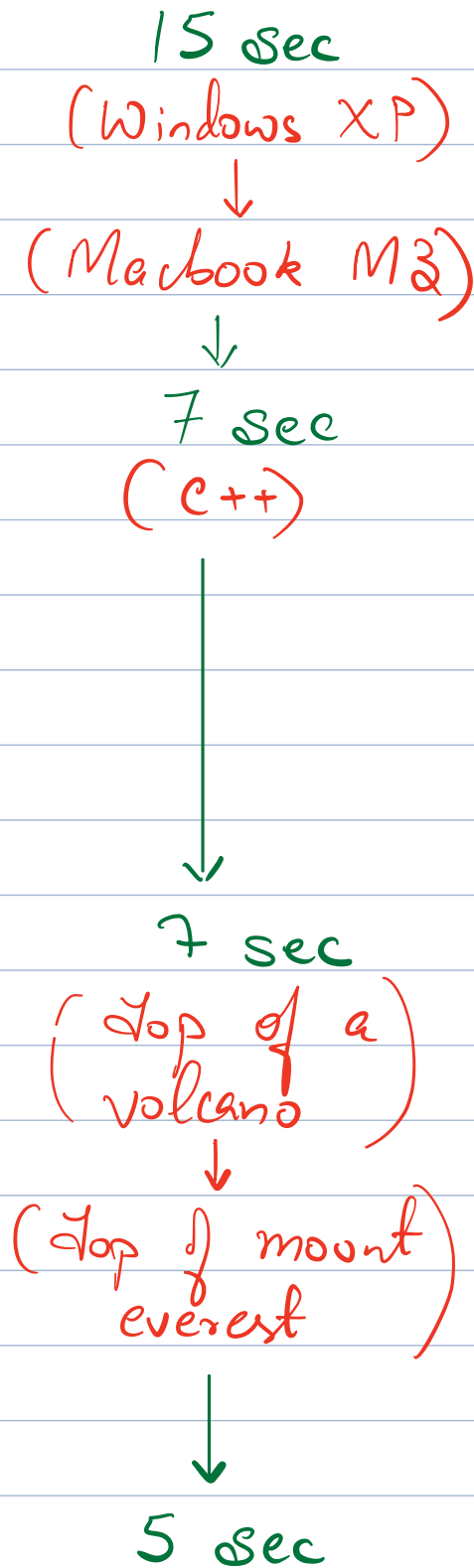
Ans = N + M

How to compare two algo

Build your own sorting algo.

Shouti
(Awesome Sort)

Lohith
(Cool Sort)



Copied from Sarika

Execution Time is not a good factor to compare 2 Algo.



Number of iterations.

$$\text{Sqrt} (10^9)$$

$$\sqrt{10 \times 10^8}$$

$$= (\sqrt{10}) \times 10^4$$

$$(3. \dots) \times 10^4$$