Paridhi Gupta

2+ years

Instructor — Developer

MCA from IGDTUW

3+ years of teaching experience - DSA.

Que 1. Given N, count all the factors of N [N>=1]

What is a factor?

↳ x → x should completely divide N.

N % x = 0

N = 10

1, 2, 5, 10

4 factors

N = 36

1, 2, 3, 4, 6, 9, 12, 18, 36

9 factors.

To: Everyone

Type message

Simple Approach / Brute Approach

```
int factors(N) {
    int count = 0
    for(int i = 1 ; i <= N ; i++ ) {
        if ( N % i == 0) {
            count++ ;
        }
    }
    return count
}
```

Iterations → no. of time the loop runs.

$i \rightarrow [1 \quad N]$

N iterations

General Assumption

10^8 iterations in 1 sec

$N \simeq 10^9$

10^8 iterations → 1 sec

10^9 " → $\frac{1}{10^8} \times 10^9 = 10$ sec

5 pencils → 10 Rs

25 pencils → $\left(\frac{10}{5}\right) \times 25$

= 50 Rs.

$N \simeq 10^{18}$

10^8 iterations → 1 sec

10^{18} iterations → $\frac{1}{10^8} \times 10^{18} = 10^{10}$ sec

$\simeq$ 317 years.

You → kids → grand kids → 4th → 5th

Optimisation

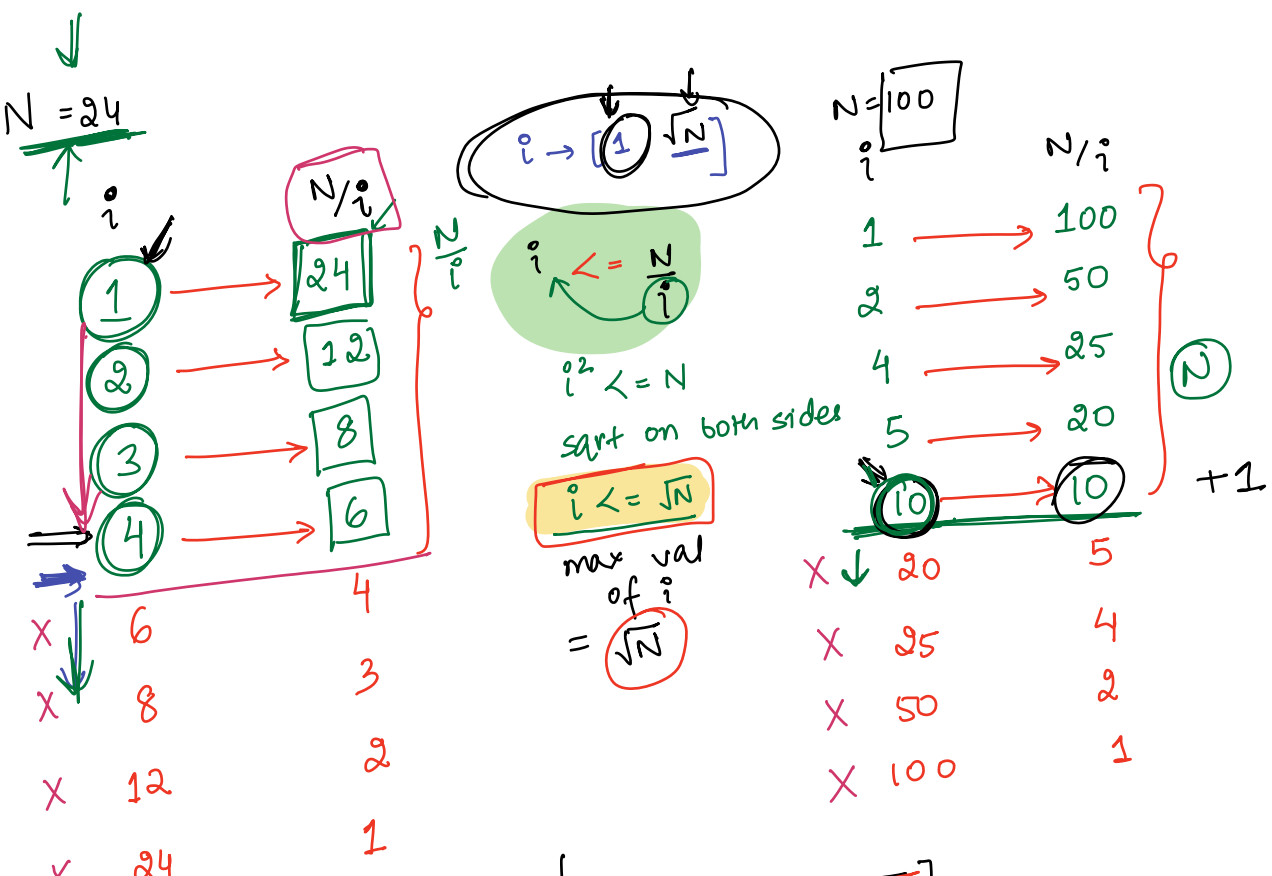
Given a, b, N → +ve nos.

given $a * b = N$, $b = N/a$

$\{a, b\}$ are the factors of N

$\{a, \frac{N}{a}\}$ are the factors of N

factors appear in pairs



Relational Operators

<, >, <=, >=, !=, ==

i → [1 √N]

√N iterations

Optimised code

```
int factors ( int N ) {
    int count = 0
    for ( int i = 1 ; i*i <= N ; i++ ){
        if ( N % i == 0) {
            if ( i == N/i ) {
                count = count + 1
            }
            else {
                count = count + 2
            }
        }
    }
    return count
}
```

$i = 10$

$N = 100$

$\frac{N}{i} \rightarrow \frac{100}{10} = 10$

$i * i <= N$

$i <= \sqrt{N}$

$N = 25$

$i = 5$

i	N/i
5	25/5 = 5

$N \leq 10^{18}$

$\sqrt{N} = 10^9 = 10s$

Que 2. Given a no. N, check whether it's prime or not.

prime no. : has exactly 2 different factors

21 is not a prime because it has factors other than itself.

5 → 1 & 5 only

2 → 1 & 2 only

15 → has factors other than 1 & 15.

(N) → counting no. of factors

⇓

N is prime or not

```
if (count == 2) {
    return true
}
return false
```

```
int factors (int N) {
    int count = 0
    for (int i = 1 ; i <= √N ; i++) {
        if (N % i == 0) {
            if (i == N/i) {
                count = count + 1
            }
            else {
                count = count + 2
            }
        }
    }
```

$i * i <= N$

N = 12

count = 6

```
if (count == 2){
    return true
}
return false
}
```

HW

Try it on the code

Sum of first N natural nos.

$\rightarrow \frac{N(N+1)}{2}$

Aditya

$$S = 1 + 2 + 3 + 4 + \ldots + 98 + 99 + 100$$

$$S = 100 + 99 + 98 + 97 + \ldots + 3 + 2 + 1$$

$$2S = 101 + 101 + 101 + 101 + \ldots + 101 + 101 + 101$$

$$2S = 100 \times 101$$

$$S = \frac{100 \times 101}{2}$$

1 — N

$S = 1 + 2 + 3 + 4 + \cdots + (N-3) + (N-2) + (N-1) + (N)$

$S = N + (N-1) + (N-2) + (N-3) + \cdots + 4 + 3 + 2 + 1$

$2S = (N+1) + (N+1) + (N+1) + (N+1) + \cdots + (N+1) + (N+1) + (N+1) + (N+1)$

$2S = N \times (N+1)$

$S = \frac{N(N+1)}{2}$

Gauss

Q. sum of first N whole nos.

0, 1, 2, ... N-1

[0 N-1] → N nos.

sum of first N-1 natural nos.

$S = \frac{N \times (N+1)}{2}$

$= \frac{(N-1) \times (N)}{2}$

whole starts from 0

[1 N] → N

[0 N] → N+1

[1 N-1] → N-1

[0 N-1] → N

first 2 whole nos. {0, 1}

first 3 whole nos. {0, 1, 2}

first N whole nos. {0, 1, .. N-1}

Que. Given N, tell no. of times we need to divide N by 2 till it reaches 1.

$N = 2 : \frac{2}{2} \to 1 : 1$

$2^1 \to \log_2 2 = 1$

$N = 4 : \frac{4}{2} \to \frac{2}{2} \to 1 : 2$

$2^2 \to \log_2 4 = 2$

$N = 8 : \frac{8}{2} \to \frac{4}{2} \to \frac{2}{2} \to 1 : 3$

$2^3 \to \log_2 8 = 3$

$N = 15 : \frac{15}{2} \to \frac{7}{2} \to \frac{3}{2} \to 1 : 3$

$\log_2 15 = 3.9$

$N = 16 : \frac{16}{2} \to \frac{8}{2} \to \frac{4}{2} \to \frac{2}{2} \to 1 : 4$

$2^4 \to \log_2 16 = 4$

$N = 27 : \frac{27}{2} \to \frac{13}{2} \to \frac{6}{2} \to \frac{3}{2} \to 1 : 4$

$\log_2 27 = 4.7$

$N = 50 \xrightarrow{/2} 1$

steps $\Rightarrow \log_2 50 = 5.64$

$= 5$

$\frac{50}{2} \rightarrow \frac{25}{2} \rightarrow \frac{12}{2} \rightarrow \frac{6}{2} \rightarrow \frac{3}{2} \rightarrow 1$

$\log_2 N$

$\log_2 2^5 = 5$

$\log_5 5^{10} = 10$

$\log_7 7^3 = 3$

$\log_a a^x = x$

Amazon MCQ question

Given a perfect square ($N >= 1$), find square root of N.

$N = 25 \rightarrow 5$

$N = 100 \rightarrow 10$

$N = 40$ x invalid i/p

```
int sqrt(N) {
    for(i = 1; i <= N; i++) {
        if (i * i == N) {
            return i
        }
    }
}
```

N = 36

i	1	2	3	4	5	6
	X	X	X	X	X	return 6

N = 25

i →

1	2	3	4	5
X	X	X	X	return 5

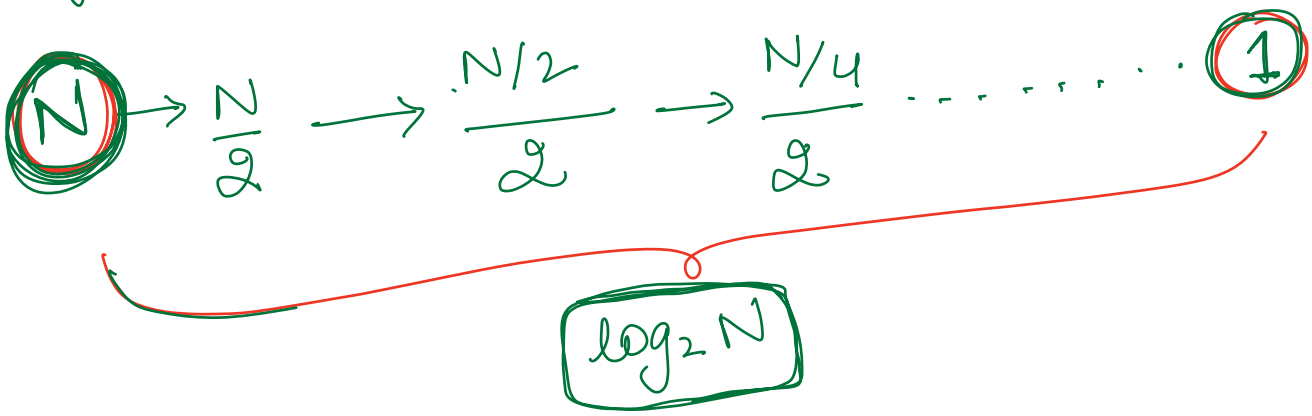
Don't worry, we'll all scenarios that we're not taking right no in coming up sessions.

[1 N]

N=100 [1 100]

	mid	
[1 100]	50	50 x 50 > 100 [51, 52, 53, ... 100]
[1 49]	25	25 x 25 > 100 [26, 27, 28, ... 49]
[1 24]	12	12 x 12 > 100 [13, 14, ... 24]
[1 11]	6	6 x 6 < 100 [1, 2, ... 5]
[7 11]	9	9 x 9 < 100 [7, 8]
[10 11]	10	10 x 10 = 100

Ignored one of the half everytime



$\sqrt{N}$ $\log_2 N$

$N = 2^{30}$ 2^{15} $\log_2 2^{30} = 30$

$N = 2^{60}$ 2^{30} $\log_2 2^{60} = 60$

Intermediate Content

Time complexity → 2 sessions

Arrays → 6 sessions

1) Intro
2) Prefix Sum
3) Carry Fwd
4) Subarray / Sliding window
5) 2D matrix
6) Interview Problems.

Bit manip → 2 sessions

Maths & Arrays : 2 sessions

Sorting / Strings / Hashmap : 4

Recursion : 3 session

subset / subsequence

linked list Basics.