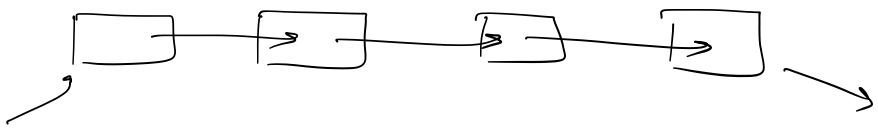
1) Why problem solving is important.
2) Good problem solver
3) Examples
4) Doubts

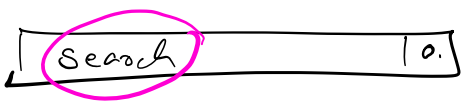
ML / DS Backend Fr..

20 years ( )



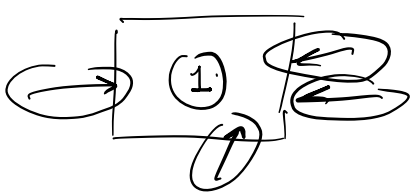
Amazon

1) www.amazon.in ⇒ Availability

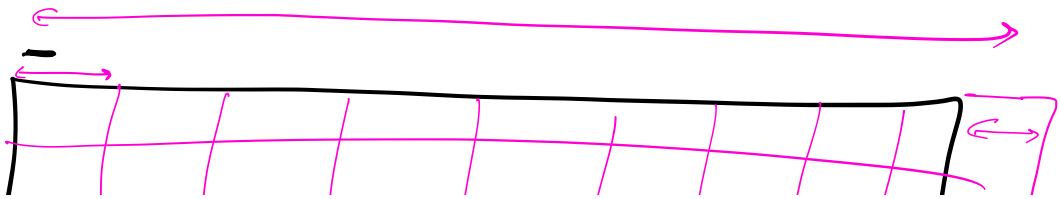
2) 

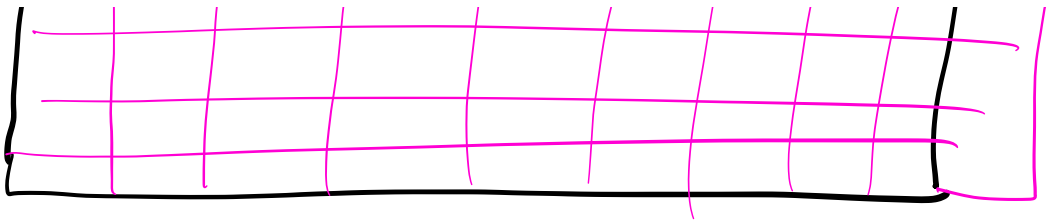
⇒ suggestions
fast / optimal

3) Checkout ⇒ concurrency



Observation





500 matchsticks

2) Germany (4/5th standard)

Gauss

$S: \quad 1 + 2 + 3 + 4 \ldots \ldots \ldots 97 + 98 + 99 + 100$

101 (1 + 100)

101 (2 + 99)

101 (3 + 98)

$S: \quad 1 + 2 + 3 + 4 \ldots \ldots \ldots 97 + 98 + 99 + 100$

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

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

100 terms

$2S = 100 \times (101)$

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

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

Factors

x is a factor of N?

10 : 1, 2, 5, 10 $\quad N \% x = 0$

24 : 1, 2, 3, 4, 6, 8, 12, 24

Code

$[1, N]$

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

Iteration

$10^9 = 10 \times (10^8)$

Assumption : $1 \text{ sec} = 10^8$ iteration

$N = 10^9 \Rightarrow 10^9$ iterations $\Rightarrow$ 10 seconds

$N = 10^{18} \Rightarrow 10^{18}$ iterations $\Rightarrow 10^{10}$ seconds

$\frac{317}{2}$ years

$1 \text{ sec} = 10^8$ iterations

1 iteration $=$ $\frac{1}{10^8}$ sec

10^{18} iterations $= 10^{18} \times \left(\frac{1}{10^8}\text{ sec}\right)$

$10^9 = \frac{10^9}{10^8}$ sec $\approx$ 1 sec.

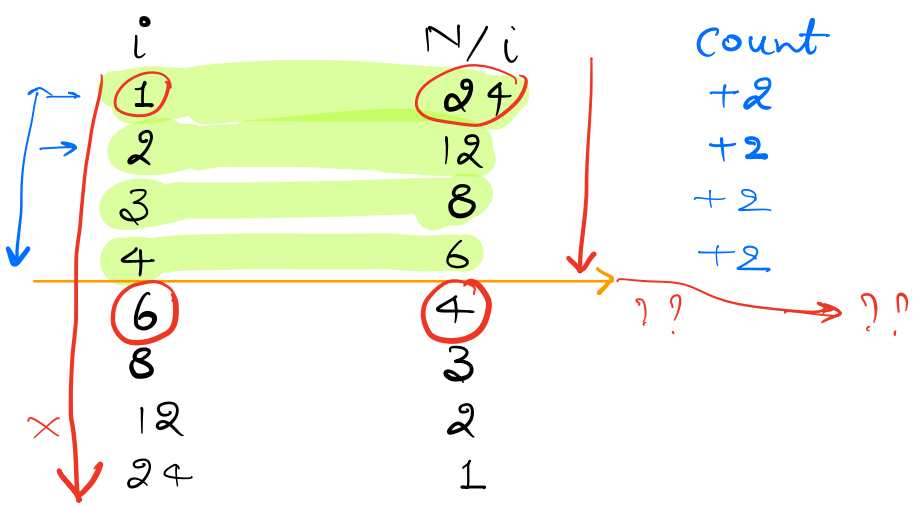
3 numbers

Given $\begin{cases} i, j, N \Rightarrow \text{integers} \\ i \times j = N \Rightarrow \text{Both } i \text{ \& } j \text{ are factors of } N \end{cases}$

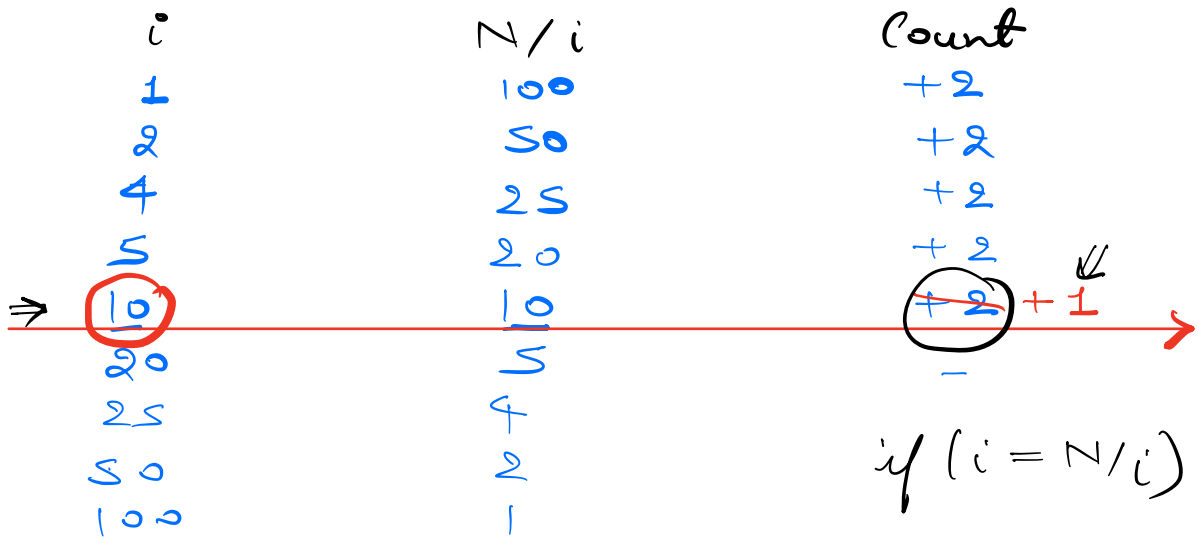
$j = N/i$

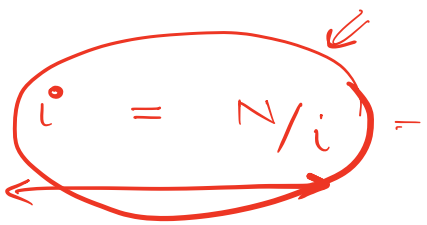
If i is a factor of N,
N/i is also a factor.

$N = 24$

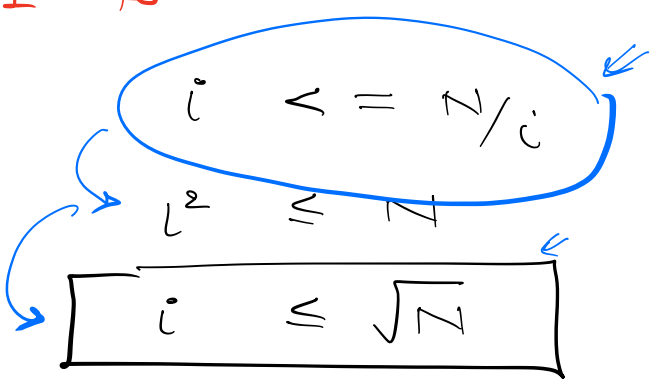


$N = 100$





1 to



$[1, \sqrt{N}]$

$N = 24$

```
int x = 24;
int y = sqrt(x);
```

$N = 24$

$[1, 4]$

print (y);

$\sqrt{24} \simeq [4.8] = 4$

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

$N = 10^{10} \Rightarrow \sqrt{10^{10}} = 10^5$ iterations

$N = 10^{18} \Rightarrow \sqrt{10^{18}} = 10^9$ iterations

$= 10$ seconds

Given N, no. of times we need to divide it by 2 to reduce until it reaches 1

$=$ $\log_2 N$

$\log_a a^m = m$

int x = 8	8	11	13
int y = x/2 ⇒	4	5	6

N: 2^1 : $2 \rightarrow 1$ $= \log_2 2^1 = 1$

2^2 : $4 \rightarrow 2 \rightarrow 1$ $\log_2 2^2 = 2$

2^3 : $8 \rightarrow 4 \rightarrow 2 \rightarrow 1$: $\log_2 2^3 = 3$

$9 \xrightarrow[/2]{} 4 \xrightarrow[/2]{} 2 \xrightarrow[/2]{} 1$: $\log_2 9 : 3.(1....)$

2^4 : $16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ $\log_2 2^4 = 4$

Q Given a perfect square N.

Find the sqrt (N)

I/P	O/P
49	7
100	10
50	⇒ Not a valid input

$[1, 2, 3, 4, \ldots\ldots \sqrt{N}]$

$N = 36$

i : 1 , 2 , 3 , 4 , 5 , 6 , 7

1 ✗ , 2 ✗ , 3 ✗ , 4 ✗ , 5 ✗ , 6 ✓ , 7

$i^2 = i \times i == N$

Pseudo Code

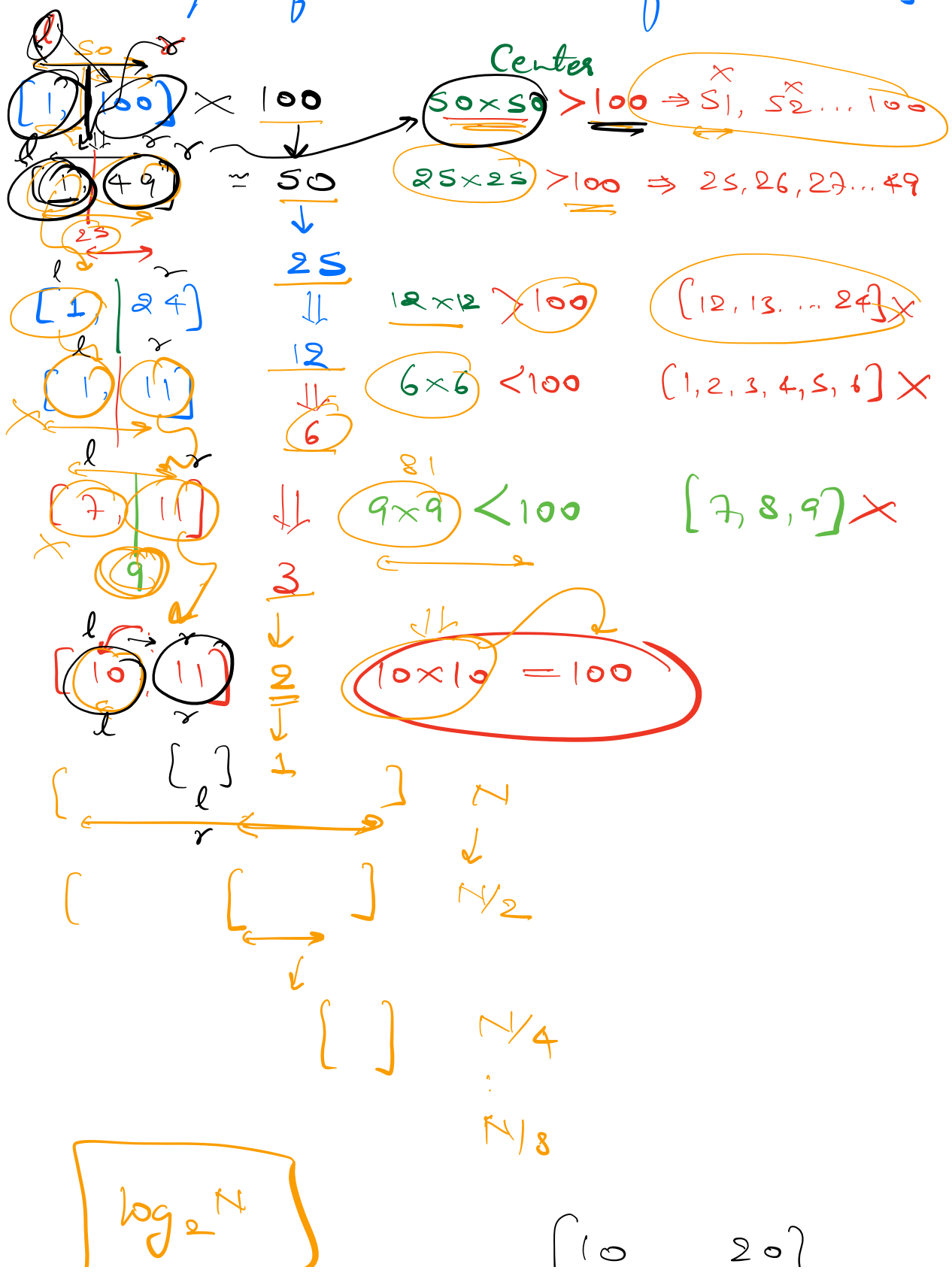
```
sqrt (N) {
    for (i=1; i<=N; i++) {
        if (i*i == N) {
            return i;
        }
    }
}
```

$i = \sqrt{N}$ ⇒

$N \Rightarrow \sqrt{N}$ iterations

$N = 100$

sqrt of 100 will lie from [1 to 100]



$\frac{10+20}{2} = 15$

$\sqrt{N} > \log_2 N$

Pseudo code

```
l = 1
r = N;

while ( l <= r ){
    mid = (l+r)/2 ;
    if ( mid*mid == N) {
        return mid;
    }
    else if ( mid*mid > N) {
        r = mid - 1;
    }
    else {
        l = mid + 1;
    }
}
```

N = 36	l	r	mid
	1	36	18
	1	17	9
	1	8	4

5 8 7 6

5 6 5 25<31

6 6

7 6

9