Agenda:

1) Why problem solving is important?
2) Key Ingredient of a good problem solver
3) Examples / Problems using the key ingredient
4) Importance of Data structures & Algo.

How to become an impactful Engineer?

Ex1 : Problem of communication & staying connected

FB, Messenger, whatsapp

Ex2 : Online booking tools

Ex3 : Online Shopping

One skill to become Impactful ?

Frontend? Backend? ML / DS

$\boxed{\text{Problem Solving}}$ ✓

Amazon (Web Application)

1) Go to www.amazon.com

2) Type in the search bar

3) Select item, checkout, Payment

4) Delivery

Step1 :

-> Amazon would have gone down very few times

-> the website should be available 100% of time.

Problem of availability. [Not an easy Problem]

-> No Down time.

99.9
99.99
999.

Step2: Type in the search

Challenges / Problems :

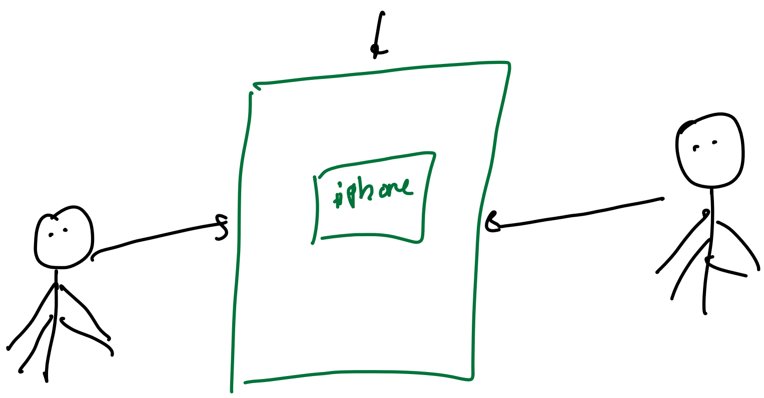
1) show Autosuggestions with a particular prefix as fast as possible

2) Searching should be very fast

Not an easy Task

Step3: select item, checkout, payment

Anything related to transactions : security issues / concerns



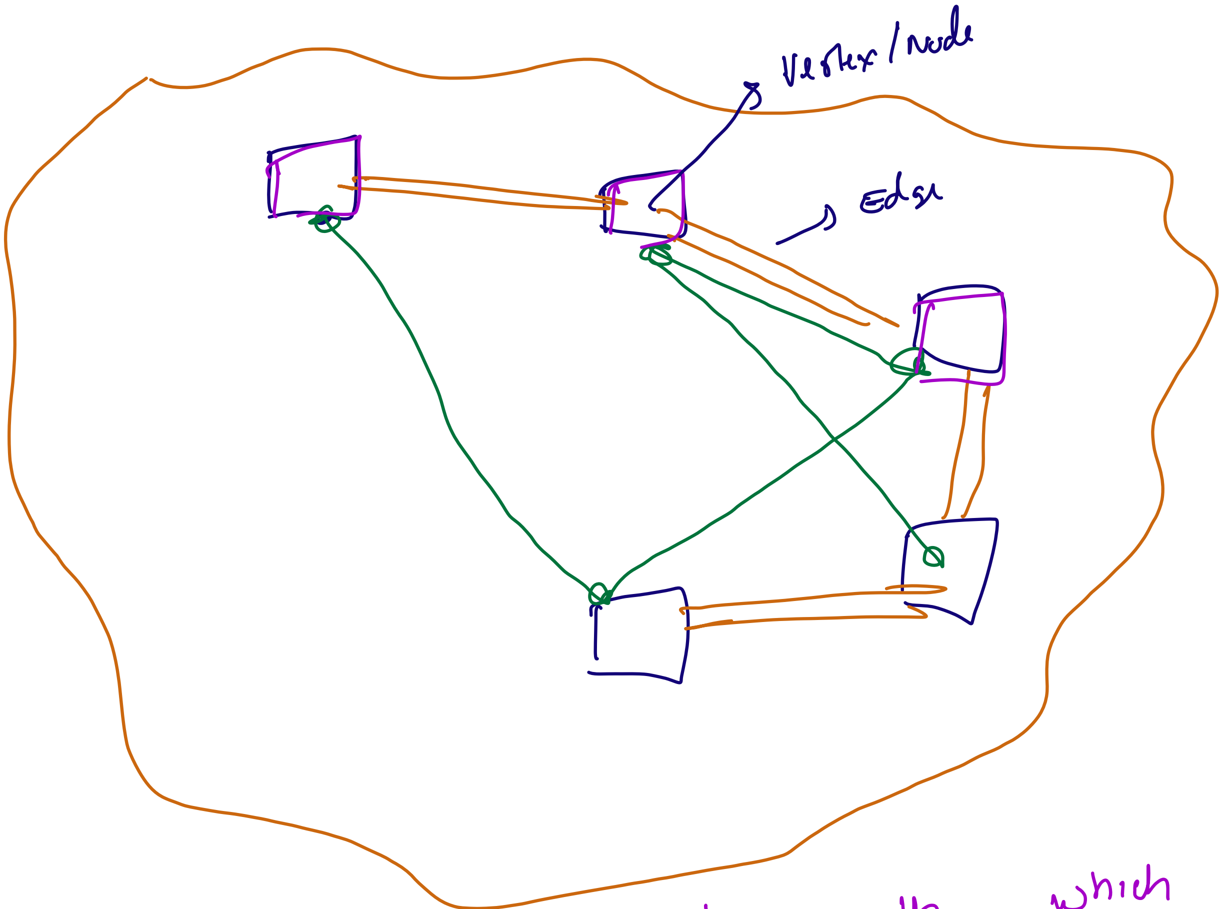
→ Buying / Booking type of systems

Problem of Concurrency

Step 4: Delivery

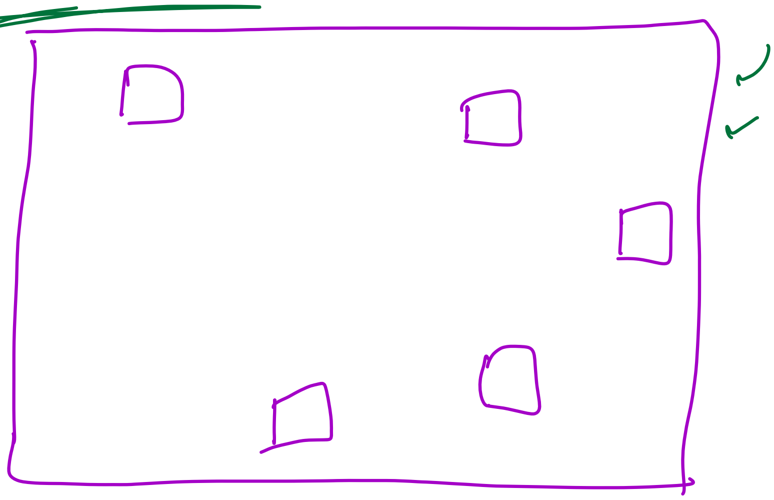
→ Use least amount of resources

→ Fuel, Delivery partners .....etc

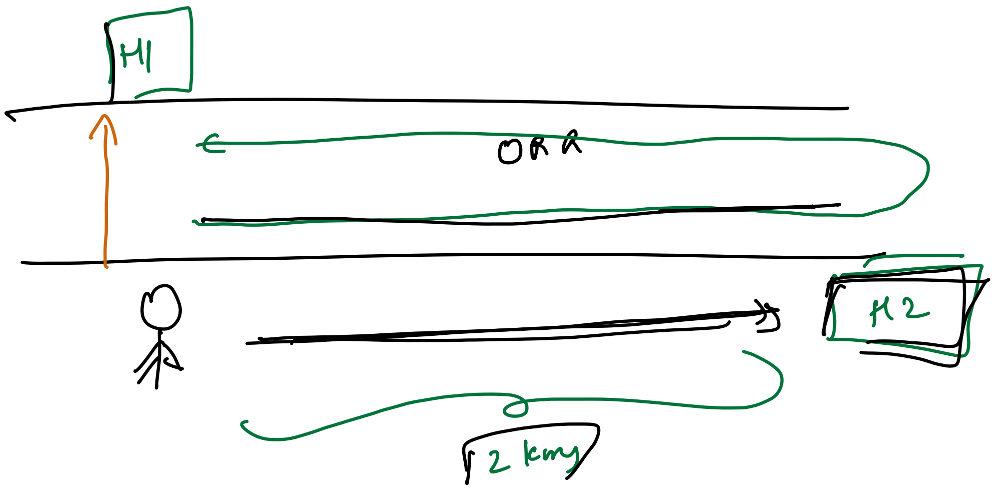


Graph Problem. Find the path which consumes least amount of resource (distance)

Expected date of Delivery



Tons of other factors

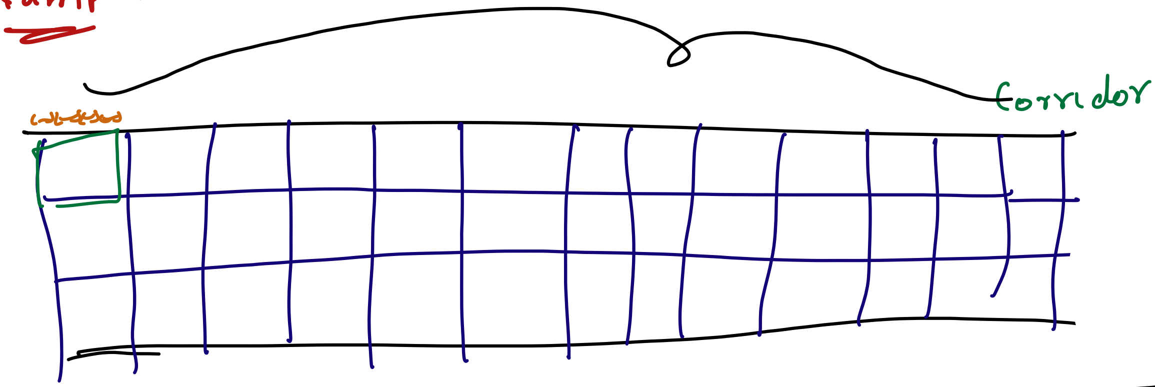


Distance is not the only criteria.

Traffic, City conditions, weather, vehicle breakdowns, 2/4 wheeler etc.

$\boxed{\text{Problem Solving}}$

Example1.



Very simple observatory

14th class

Example2: Germany

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

101

101

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

101

101

$a + b = b + a$

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

$$(+)\quad S = 100 + 99 + 98 + 97 \ldots + 4 + 3 + 2 + 1$$

$$2S = \underbrace{101 + 101 + 101 + 101 + 101 + \ldots 101 + 101 + 101}_{100 \text{ times}}$$

$$2S = 101 \times 100$$

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

$\boxed{\text{Gausse}}$ ✓

$$S = 1 + 2 + 3 + \dots \dots + N-2 + N-1 + N$$
$$S = N + N-1 + N-2 + \dots \dots + 3 + 2 + 1$$

$$2S = \underbrace{(N+1) + (N+1) + (N+1) \quad (N+1) + (N+1) + (N+1)}_{N \text{ times}}$$

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

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

Key Ingredient

Making Observation

Prime Numbers

-) Positive Numbers which have exactly 2 distinct factors

-) 1 and it self

2, 3, 5, 7, 11, 13 . . .

15 => 1, 3, 5, 15 => $\boxed{4}$

12 => 1, 2, 3, 4, 6, 12 => 6

9 => 1, 3, 9 => 3

23 => 1, 2 => $\boxed{2}$

1 => 1 [only 1 factor]

Smallest Prime : 2

$$24 = \underbrace{\overset{(1)}{1}, \overset{(2)}{2}, \overset{(3)}{3}, \overset{(4)}{4}, \overset{(5)}{6}, \overset{(6)}{8}, \overset{(7)}{12}, \overset{(8)}{24}} \Rightarrow 8 \text{ factors}$$

-) 3 is a factor of 24,

$$\frac{24}{3} = 8$$

remainder = 0

24 / 3 has 0 remainder.

How to check if 'i' is a factor of N?

⇒ Remainder of N / i has to be 0.

% ⇒ Module → Finds the remainder.

Quotient ⇒ N/i

Remainder ⇒ N % i

Eg:

34 / 5 ⇒ 6 , 34 % 5 = 4

34 = 6 × 5 + 4

17/3 ⇒ $\boxed{5.6 \ldots}$, 17 % 3 = 2

≈ 5

3) 17 (5.6
15
20

29 / 7 ⇒ 4 , 29 % 7 = 1

35 / 5 = 7

$\boxed{35 \% 5 = 0}$

$\boxed{N \% i}$

Question: check if a number is Prime

Approach 1 : Impl 1

```
int count = 0;
for(i=1; i<=N; i++){
    if(N%i == 0){
        count++;
    }
}
if(count == 2){
    "PRIME"
}
else{
    "NOT PRIME");
}
```

$i = 1, 2, 3, 4 \ldots N$

N iterations

1st Factor = 1

2nd Factor = N

Impl2 :

[1, N] => 2 factors

[2, N-1] => 0 factors

$i = [2 \ldots N-1]$

N = 2

i = 2

i = 3

```
{
for(i=2; i<=N-1; i++){
    if( N%i == 0){
        return "NOT PRIME";
    }
}
return "PRIME";
```

N%i == 0

N = 3

i = 2

3%2 == 0?

N = 2 -> Prime

N = 3 -> Prime

N = 0, 1

```
if(N <= 1){
    "NOT PRIME"
}
```

Approach2 :

3 is a factor of 24

$\frac{24}{\boxed{3}} = 8 \Rightarrow \boxed{\frac{24}{8}} = \boxed{3}$

$\frac{24}{3}$

$4 \Rightarrow \frac{24}{4}$

→ if 4 is a factor, 6

→ if 2 is a factor, 12

$a * b = N$ (a, b are factors) $\Rightarrow b = \frac{N}{a}$

if $\boxed{\text{'a'}}$ is one Factor, the other factor is N/a

N = 24

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

$\frac{N}{2}$

$N = 24 \Rightarrow \boxed{12}$

$\boxed{N = 100}$

	a		N/a
1st Half, $a \le \frac{N}{a}$	1	$<$	100
	2	$<$	50
	4	$<$	25
	5	$<$	20
	$\boxed{10}$	$\le$	10
2nd Half, $a > \frac{N}{a}$	20	$>$	5
	25	$>$	4
	50	$>$	2
	100	$>$	1

$\boxed{a \le \frac{N}{a}}$

$$\boxed{a \le \frac{N}{a}}$$

$$\sqrt{a^2} \le \sqrt{N}$$

$$\boxed{a \le \sqrt{N}}$$

a, b are factors

→ If $N = \boxed{a \times b}$ Definitely less

one factor should be

than or equal to $\sqrt{N}$

$$[2, \sqrt{N}]$$

→ If there are no factors, this number is def a prime

$\boxed{\text{Function}}$

$\boxed{i \le \sqrt{N}}$ $\Rightarrow$ $\boxed{i \times i \le N}$ $\boxed{2 \times 3} =$

O(1)

sqrt : O(log n)

```
for (i = 2; i*i <= N; i++) {
    if( N % i == 0) {
        " NOT PRIME "
    }
}
"PRIME"
```

sqrt(N)

N = 100

i = 2, 3, 4, 5 10

i = 2

$n^{0.5}$

O(1)

iterations = $\sqrt{N}$

O(1)

$\Rightarrow$ 1 iteration takes $\Rightarrow$ 1 ms

Approach1 : N

Approach2 : $\boxed{\sqrt{N}}$

N = 100 | 100 ms | 10 ms

N = 10^6 | 10^6 ms = 1000 s = 16.66 mins | $\sqrt{10^6}$ ms = 10^3 ms = $\boxed{1 s}$

N = 10^{18} = 10^{18} ms = 31687646 years | $\sqrt{10^{18}}$ = 10^9 ms = $\boxed{11 \text{ days}}$

After 6 mins

1 hr more

Brush up School Maths

$\rightarrow \sqrt{N}$

$\boxed{N = 10^{18}} \Rightarrow \sqrt{10^{18}}$

$\sqrt{x} = x^{1/2}$

$\sqrt{10^{18}} = \boxed{(10^{18})^{1/2}}$

$((10)^x)^y = 10^{x \cdot y}$

$= 10^{18 \times \frac{1}{2}} = \boxed{10^9}$

Question: Given a number, how many times we need to divide by 2 to reduce it to 1?

	N		steps
2^1	2	$2 \xrightarrow{/2} 1$	1
2^2	4	$4 \xrightarrow{/2} 2 \xrightarrow{/2} 1$	2
2^3	8	$8 \xrightarrow{/2} 4 \xrightarrow{/2} 2 \xrightarrow{/2} 1$	3
2^4	16	$16 \xrightarrow{/2} 8 \xrightarrow{/2} 4 \xrightarrow{/2} 2 \xrightarrow{/2} 1$	4
	15	$15 \xrightarrow{/2} 7 \xrightarrow{/2} 3 \xrightarrow{/2} 1$	$\boxed{3}$ (8)
	27	$27 \xrightarrow{/2} 13 \xrightarrow{/2} 6 \xrightarrow{/2} 3 \xrightarrow{/2} 1$	4 (16)
	31	$31 \xrightarrow{/2} 15 \xrightarrow{/2} 7 \xrightarrow{/2} 3 \xrightarrow{/2} 1$	4 (16)
2^5	= 32	$32 \to 16 \to 8 \to 4 \to 2 \to 1$	5

Approach 1:

If $\boxed{N = 2^K}$ K is the answer

$2^1, 2^2, 2^3, 2^4 \ldots$

$40 : \Rightarrow 32 \Rightarrow 5$

$57 : \Rightarrow 32 \Rightarrow 5$

$\underbrace{[32 \ldots\ldots 63]}_{5} \Rightarrow 32 = 2^5$

$\underbrace{[64, \ldots\ldots 127]}_{6} \Rightarrow 64 = 2^6$

In mathematics, this is called $\boxed{\log_2^N}$

$\log_2^N \Rightarrow$ No. of times N has to be divided by 2 to make it 1.

$\log_y^x =$ No. of times x has to be divided by y to make it 1.

$a^b = \frac{\cancel{a} \times \cancel{a} \times a \ldots\ldots \; b \text{ times}}{a \times a}$

$\log_a^{a^b} =$

→ Divide exactly 'b' times to make it 1

$\log_a a^4 =$ $\boxed{a \times a \times a \times a} = a^4$

$a^4 \xrightarrow{/a} a^3 \xrightarrow{/a} a^2 \xrightarrow{/a} a \xrightarrow{/a} 1$

$(\boxed{10 : \times \times \times})$

$$\boxed{\log_a a^b = b}$$

Question: Given a perfect square, find the square root of the number

16 ⇒ 4

25 ⇒ 5

110 ⇒ [Invalid Input]

121 ⇒ 11

150 ⇒ [Invalid Input]

Approach1:

$N = i \times i$

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

N iterations

N = 100

i	i x i
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
$\boxed{10}$	100

$i = \sqrt{N}$

$Max(i) = \sqrt{N}$

• # iterations = $\boxed{sqrt(N)}$

Approach 2:

$N = 100$

$sqrt(100) =$ [1, 100]

1 iteration / 1 iter / 1 shot → Steps: { → Find middle → check condition → updating our answer space }

3 iteral

#possibilities	Step No				
N	0	[1 . . . . . . 100] = 100	50 x 50	> 100	50, 51, 52, 53 100
$\frac{N}{2}$	1	[1 . . . . . . 49] ≃ 50	25 x 25	> 100	25, 26, 27 49
$\frac{N}{2^2} = \frac{N}{4}$	2	[1 . . . . 24] ≃ 25	12 x 12	> 100	12, 13, 14 25
$\frac{N}{2^3} = \frac{N}{8}$	3	[1 . . . . 11] ≃ 12	6 x 6	< 100	6, 5, 4, 3, 2, 1
$\frac{N}{2^4}$	4	[7 . . . . . . 11] ≃ 6	9 x 9	< 100	9, 8, 7
$\frac{N}{2^5}$	5	[10, 11] ≃ 2	10 x 10 == 100		

#steps => 6

step1 step2 step3 last step

$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \, . \, . \, . \, . \, . \, . \, . \, . \; 1$

#steps = $\boxed{\log_2^N}$ steps

#

1 step => 3 operations

#steps =>

#iterations = $\log_2^N$

Approach 1

$\sqrt{N}$

$N = 2^{32} \Rightarrow \boxed{2^{16} \text{ ms}} = \boxed{10^5 \text{ Years}}$

Approach 2

$N = 2^{32}$

$\log_2^N = \log_2^{2^{32}} = \boxed{32 \text{ ms}}$

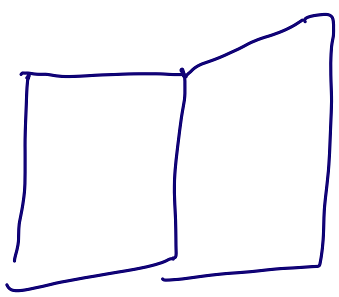
$\log_a^{a^b} = b$

Why & What Data Structures?

→ Data structures are a means of organizing and storing data in computers so that we can perform operations on the stored data more efficiently.

D. S

home



→ Random Boxes. D S

→

→ Novel

/home

Read word by word

→ Glossary / Appendix

"home": [36, 112, 236. . .]]

Hashmap Map

2:30 → 2:45

/Doubts

Undo / Redo ⇒ Stack

Auto complete ⇒ Trie

Facebook / Google Map ⇒ Graph

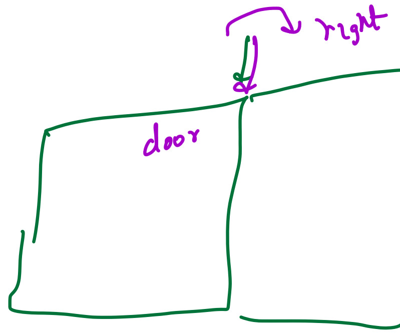
Comment

Vamsi

home

A

h.



shoes

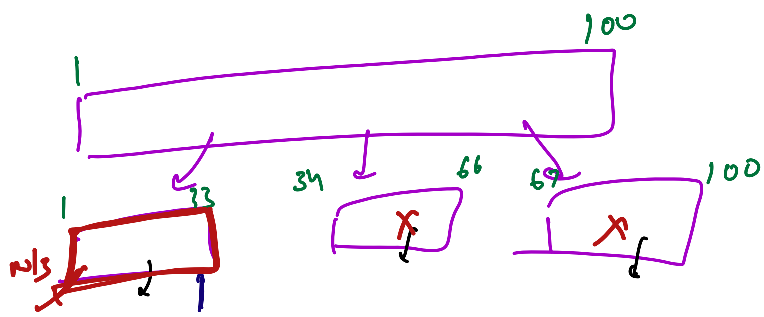
A, B, C, D E . . . Z

1 100

33 x 33 > 100

4-5 → A, B

3-4 → Home



$N \to N/3 \to N/3^2 \to N/3^3 \; . \; . \; . \; . \; 1$

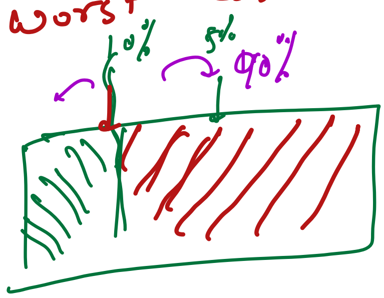
$\boxed{\log_3 N}$

$\boxed{N = 23}$ $\boxed{N = 24}$

```
for (i = 2; i ≤ N ; i++){

}
```

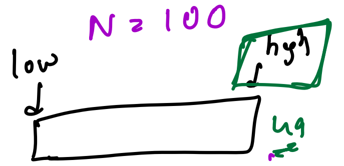
```
for (i = 2;   i*i ≤ N; i++){

}
```

→ Consider the worst case



[1 . . . . . 100]

N = 100

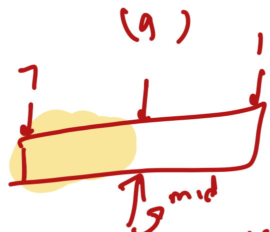


low ≤ high

low = 1   mid = 50

high = N

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



N = 100

99x < 100

Intermediate (2 months)

- Time Complexity (2 Lecture)
- Arrays / 2D Arrays (2)
- Prefix Sum / Carry (2 lecture)
- Bit Manip (3)
- Recursion → 2
- Hashing → 2
- Sorting → 1
- Basics of Linked → 1
- Basics of Stacks / Queues → 1
- Basics on Trees → 2

22-24

$\boxed{\log_2 N}$ → Math Property

9 - 12 Session

11 or 12

Apr 22 Intermediate Morning

Apr 22 US Intermediate