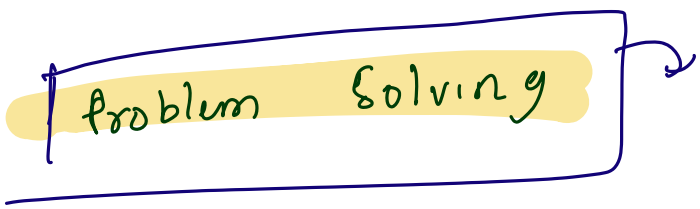
Agenda

1) Why Problem Solving is important?
2) Key Ingredient of a good Problem Solver
3) Couple of Question
4) Importance of Data structures?

How to become an Impactful Engineer?

Skill!

-> Backend Engineer!
-> Frontend Engineer
-> Java, Spring Boot
-> ML / Data scientist



Amazon

1) Go to www.amazon.com

100% available

-> Problem of availability!

Not an Easy Problem!

2) Type in the search Bar

→ 2 issues

1) Auto suggestions in very less time

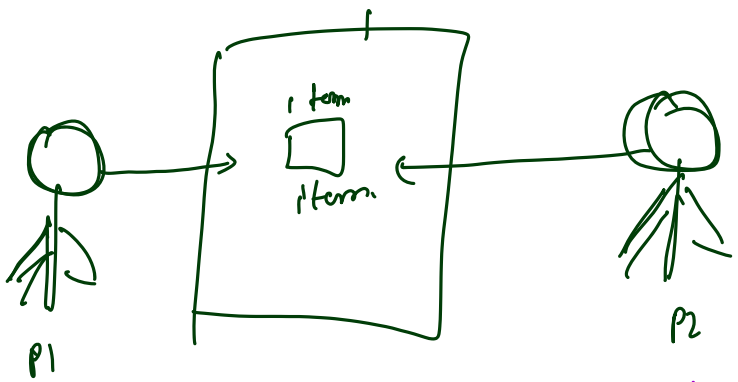
2) Searching also has to be very fast

Problem solving skills

Good amount of

3) Checkout, Add to Cart, Payment

Security Issues [Deal]



Booking / Buying type of System

Problem of Concurrency.

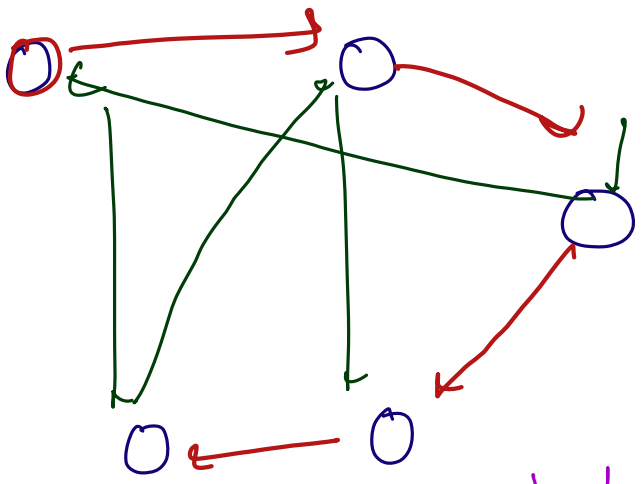
4) Delivery

→ On-time Delivery

→ Optimized Routes

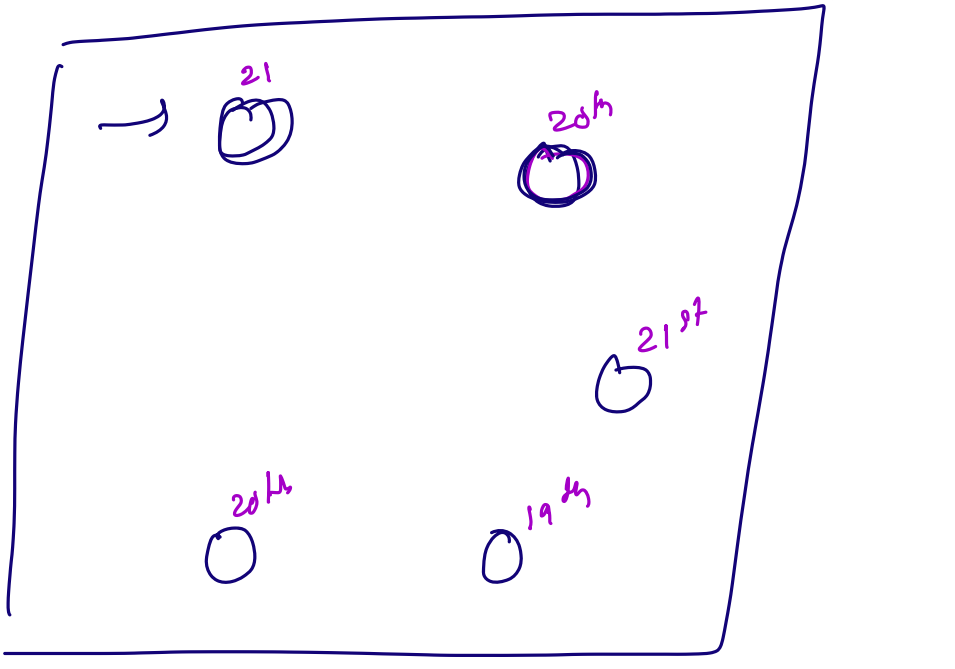
→ Delivery Partners

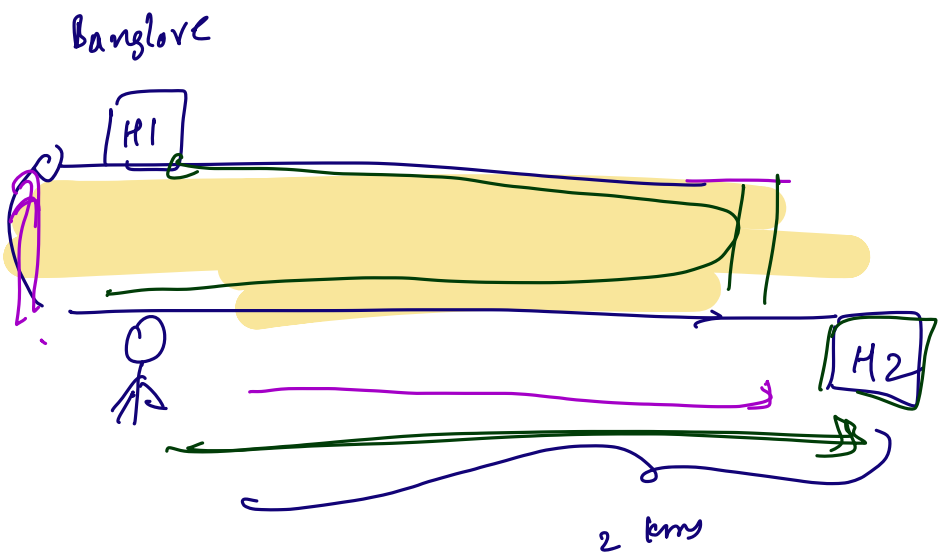
→ Fuel



Graphs: Find a path which consume least amount of resource (distance)

Expected date of Delivery





→ City Conditions

→ Traffic

→ Weather

→ 2 / 4 wheeler

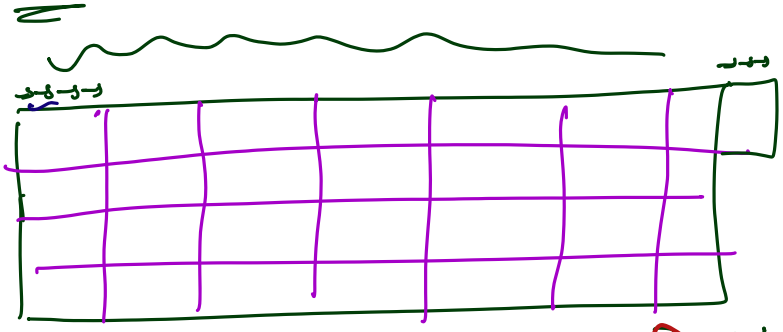
Very Tough Problem

→ Best Problem Solving

Story - 1

· Ice-breakers

match stick



corridor

[Nilgiri]

Story -2 [Germany] [4th class] 5050

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

$1+2+3+4 \ldots$

$1 + 2 + 3 + 4 + \ldots\ldots 96 + 97 + 98 + 99 + 100$

100 times

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

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

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

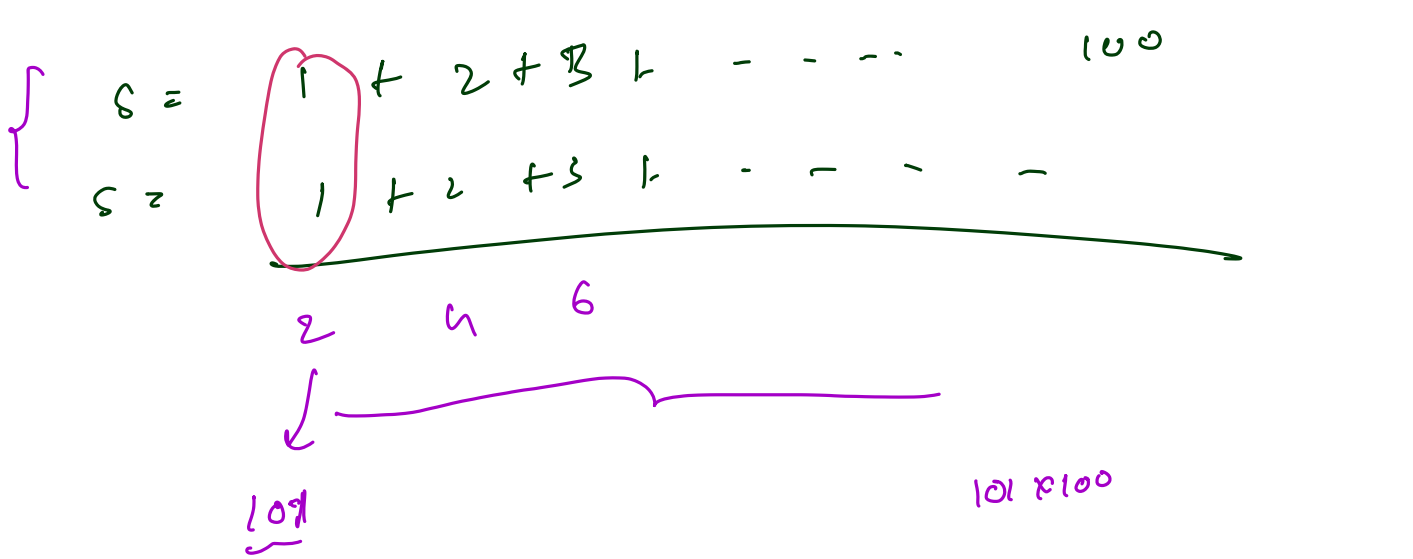
$2S = 101 \times 100$

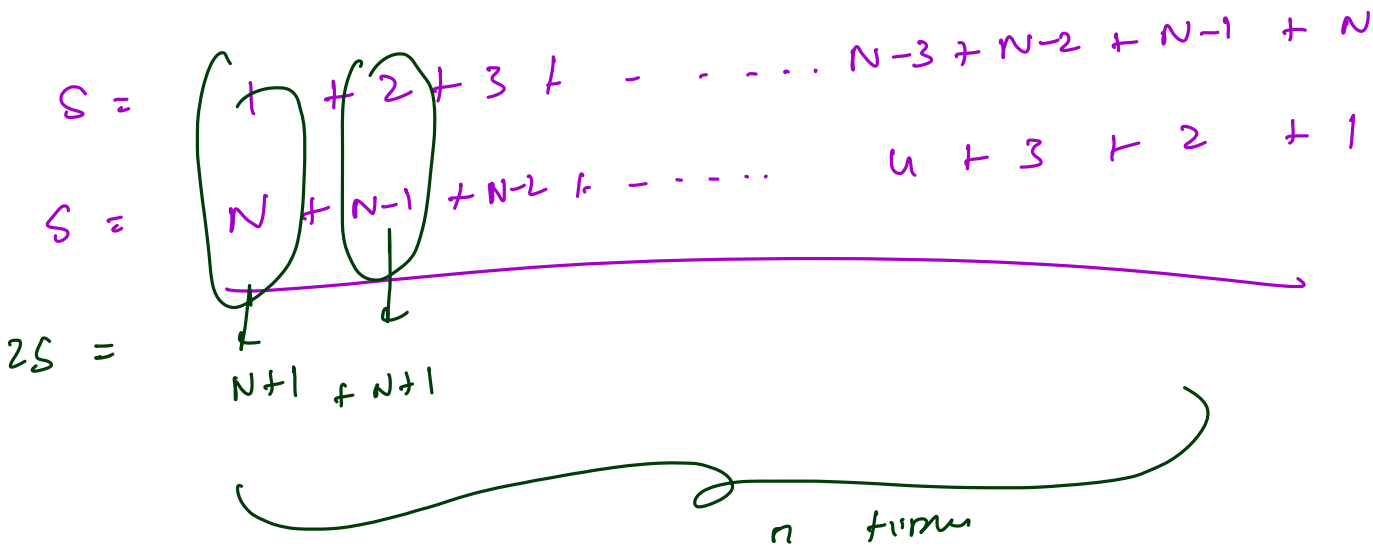
$S = \frac{101 \times 100}{2} = 5050$

Sum of First N whole numbers

$1 + 2 + 3 + \ldots\ldots N$

$N = 100 \Rightarrow 5050$





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

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

Gausse

OBSERVATIONS

→ Which we are obvious only they are told

Prime Numbers

2, 3, 5, 7, 11, 13, 17, 19, 23 - - -

All positive Numbers which have exactly 2 distinct factors (divisors) (1 and itself)

7 : 1, 7 → {2 factors} → {1, 1, 1, 1, 1, 1, 1} → 1 fac

11 : 1, 11

15 : 1, 3, 5, 15 {4 factors}

{1, 1} → {1

1 : {1} → 1 factor ✗

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

8 factors

24,

Is 3 a factor of 24 ?

% → Modulus.

24 % 3 → Remainder when 24 is divided by 3

if (24 % n == 0)

n is a factor of 24

Question: Check if a Number is Prime

Approach1:

count = 0

N = 11

i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

[1, 11]

Pseudocode

Impl1

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

i = 1, 2, 3, ... 11

N = 5

i = 1
i = 2
i = 3
i = 4
i = 5
i = 6

5 iterations

N iterations

[1, 11]

11 % 1 = 0

24, 6

(24 % 6 == 0)

i > N

25 Quiz

Impl2: N = 11

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

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

Approach 2

Observation:

3 is a factor of 24.

4 is a factor of 24?

$3 \times 8 = 24$ ($\frac{24}{3}$)

$4 \times 6 = 24$ $\frac{24}{4}$

$\Rightarrow$ If a is a factor of N,

$\frac{N}{a}$ is also a factor of N

$\rightarrow$ $\frac{24}{3} = x$ → integer.

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

$\frac{24}{x}$

Remainder: 0

$\Rightarrow$ x is a factor of 24

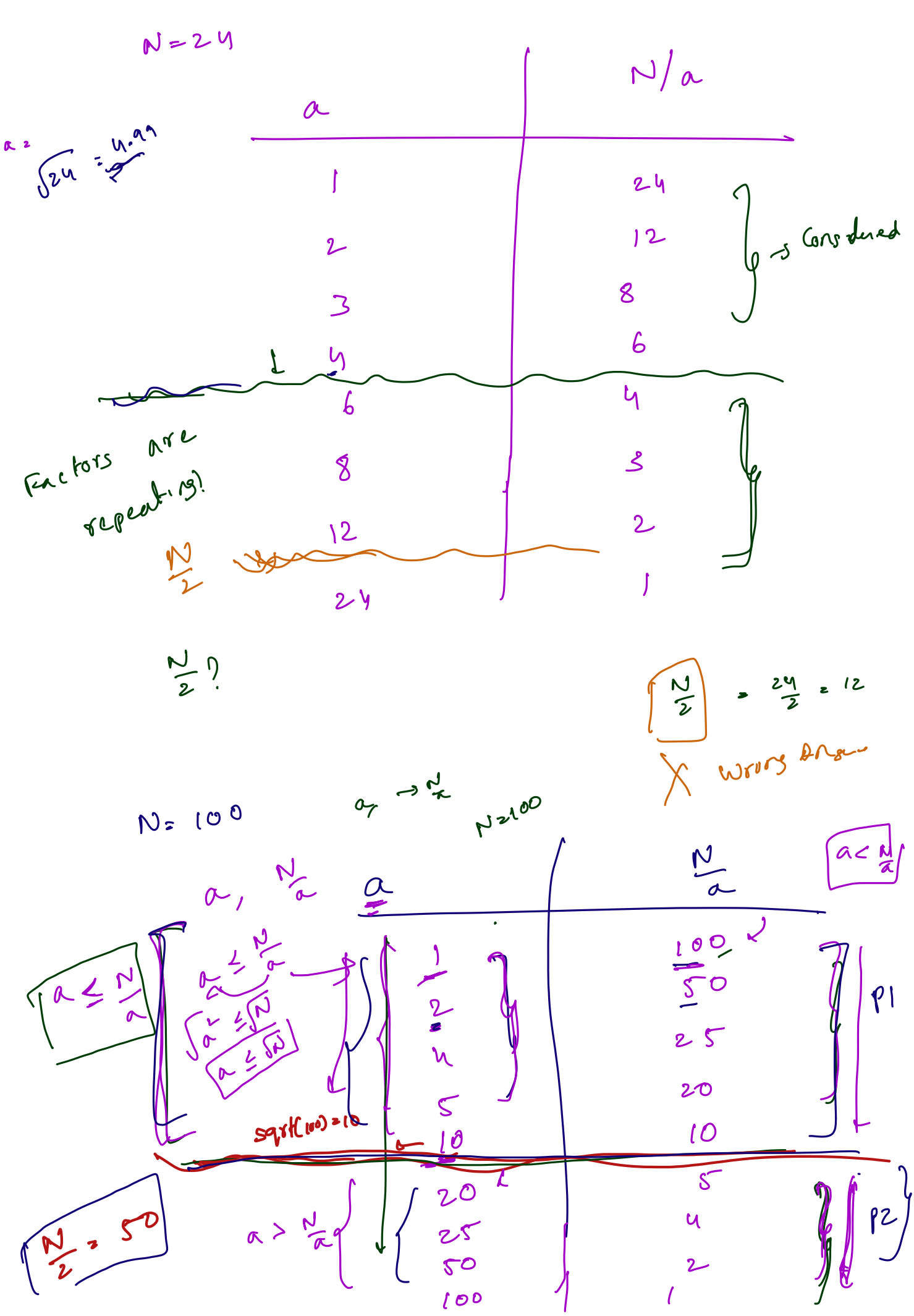
$\frac{N}{3} = \frac{24}{3}$

If $a * b = N$

(a, b) both are factors of N

$b = \frac{N}{a}$

$(a, \frac{N}{a})$ are the factors


N = 24
N/a
a
a ≤ √24 = 4.99
1
2
3
4
6
8
12
24
24
12
8
6
4
3
2
1
Considered
Factors are repeating!
N/2
N/2 ?
N/2 = 24/2 = 12
Wrong Answer
N = 100
a → N/a
N=100
a, N/a
a
N/a
a ≤ N/a
a < N/a
a ≤ √N
sqrt(100) = 10
100
50
25
20
10
5
4
2
1
P1
P2
N/2 = 50
a > N/a

Consider the factors only when

$a \leq \frac{N}{a}$

$i \leq \sqrt{N}$, for low

$a \leq \frac{N}{a}$

$a^2 \leq N$

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

$a \leq \sqrt{N}$

Sqrt(N) = $\log N$

$\log_2^{2^6} = 6$

x = sqrt(N) → 2^6

for(i = 2; i <= sqrt(N); i++)

i * i <= N

2 operations

$a = b \Rightarrow a^2 = b^2$

$a < b$, $a^2 < b^2$

$i \leq \sqrt{N}$

i * i <= N

i * 1

i + 1

1) Multiplication
2) Comparison

$i \leq \sqrt{N}$ | $i^2 \leq N$

→ If there is no factor in the range [2, $\sqrt{N}$], it is Prime Number

= Approach 2:

i = 2, ... sqrt(N)

#iterations: $\sqrt{n}$

1, N, N^2

for (i = 2; i * i <= N; i++) {

i <= sqrt(N);

IF (N % i == 0) {

"NOT PRIME"

return;

}

}

2 % 3 = 0

"PRIME":

N = 21

$\sqrt{21} = 4.5 \ldots$

a = { 2, 3, 4 }

a

2

3

N/a

N = 19

$\sqrt{19} = 4.2\ldots$

a = { 2, 3, 4 }

a	N % a
2	✗
3	✗
4	✗

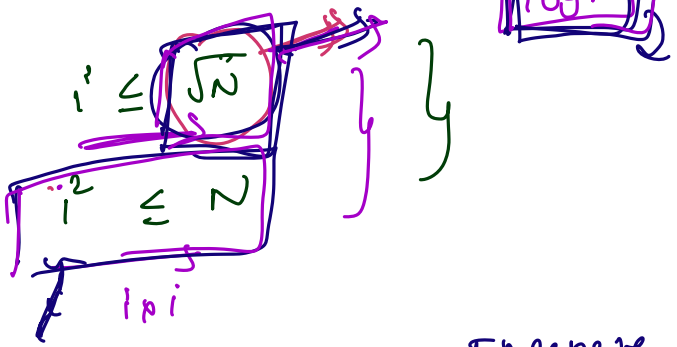
$i \le \sqrt{N}$

$(i)^2 \le (\sqrt{N})^2 \Rightarrow i \times i \le N$

O(1)

Condition 1:

Condition 2:



Easier & less Expensive operation

1 iteration ⇒ 1ms

Approach1. N

N = 100 / 100 ms

N = 10^6 / 10^6 ms = 1000 s = 16.66 min

N = 10^{18} 10^{18} sec

3168876 46 ⇒ Years

Approach2 $\sqrt{N}$

10ms

10^3 ms = 1 s

10^9 ms = 10^6 s ↓ 11 days

In Regular Computer

1 iteration ≈ 10^{-6} to 10^{-7} s

HW: 10^{18} ?

How much time will our computer take?

4-5

Question:

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

	N =	Ans.
2^1	2 $\xrightarrow{/2}$ 1	1
2^2	4 $\xrightarrow{/2}$ 2 $\xrightarrow{/2}$ 1	2
2^3	8 $\xrightarrow{/2}$ 4 $\xrightarrow{/2}$ 2 $\xrightarrow{/2}$ 1	3
2^4	16 $\rightarrow$ 8 $\rightarrow$ 4 $\rightarrow$ 2 $\rightarrow$ 1	4

8, 16

15 → 7 → 3 → 1 (1, 2, 3) — 3 steps → 8

→ 27 → 13 → 6 → 3 → 1 (1, 2, 3, 4) — 4 steps → $2^4 = 16$

35 → 17 → 8 → 4 → 2 → 1 — 5 steps → $2^5 = 32$

Observation

if ($N = 2^K$)

Answer is "K"

else

Nearest Power of 2 less than N

32 : 5

(64

[32 . . . . . 63) ⇒ 5

25

[127] → 64 ⇒ 6

[64 . . . . . 127] ⇒ 6

In mathematics this called as $\log_2 N$

→ $\log_2^N$ = No. of times I have to divide N by 2 to make it 1

$\log_x^y$ = No. of times I have to divide y by x to make it 1

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

No of times I have to divide b^a by b to make it 1

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

$a = 5$
$b = 3$

3^5 (1) 3^4 (2) 3^3 (3) 3^2 (4) 3 (5) 1

5

[Amazon]

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

N	Ans
16	4
25	5
121	11
110	[Invalid Input]
125	[Invalid Input]

→ sqrt(N) ✗

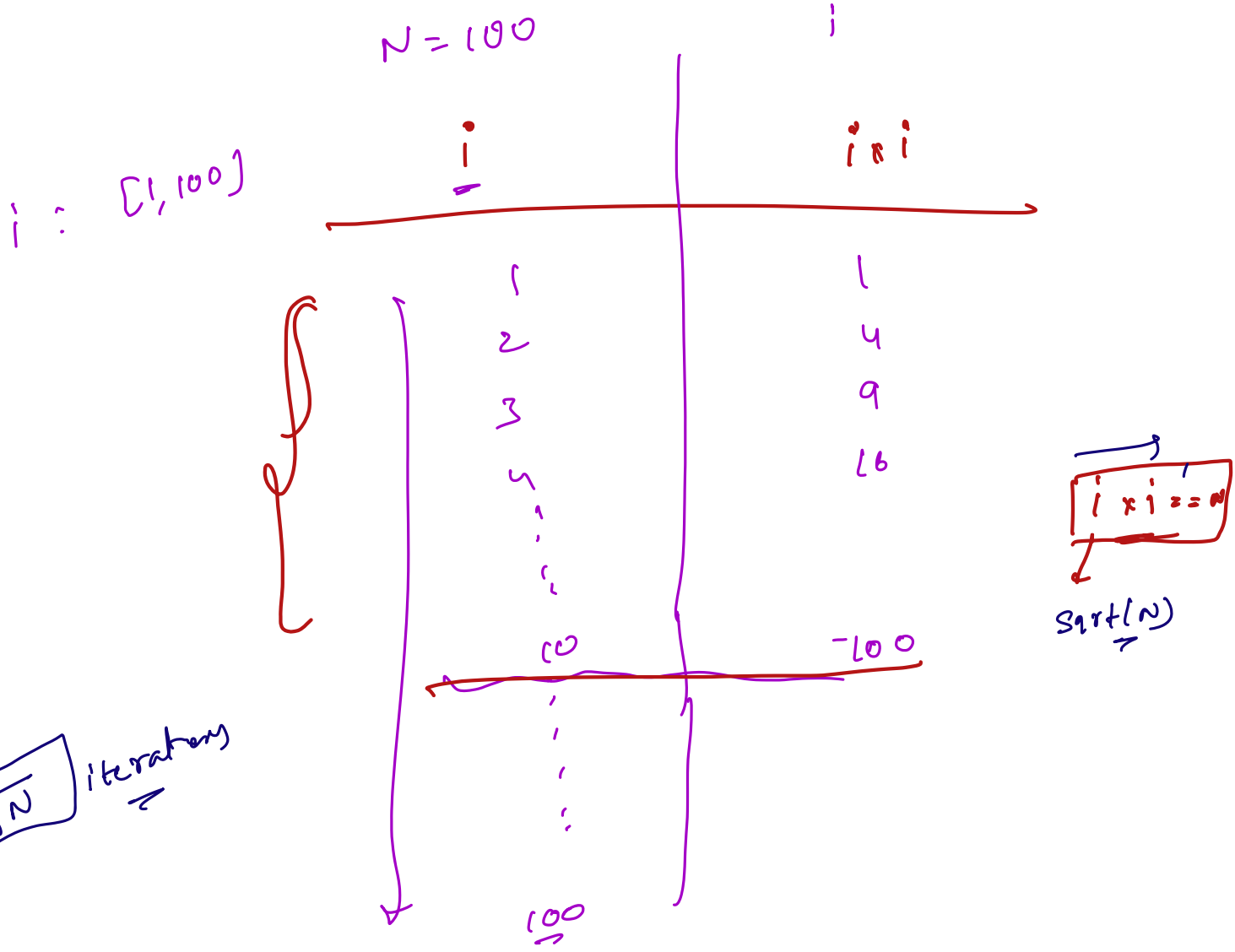
Approach1:

N is a perfect square

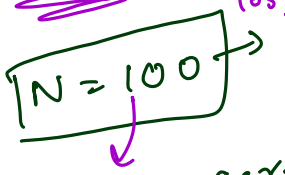
i x i == N

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

N



Approach 2:



$\log_2^{100}$

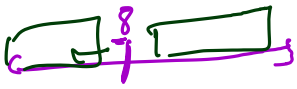
$\log_2^{36}$

$\sqrt{100} = 10$

$\sqrt{36} = 6$

$[1, N]$

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



$8 \times 8 = 64$

$\text{sqrt}(100) \geq 1$

$\text{sqrt}(N) \geq 1$

$\text{sqrt}(100) \leq 100$

$\text{sqrt}(N) \leq N$

51×51

Step $[1, 100]$

#Mid Middle

1 → [1, 2 . . . . . 100] 100 50 100 $50 \times 50 > 100$ 50, 51, 52, ... 100

$26 \times 26 > 100$

2 → [1, 2 . . . . . 50] ≤ 50 25 $\frac{100}{2}$ $25 \times 25 > 100$ 25, 26 49

3 → [1, 2 . . . . . 25] ≤ 25 12 $\frac{100}{2^2}$ $12 \times 12 > 100$ 12, 13, 14 . . . 24

5×5

4 → [1, ... 12] ≤ 12 6 $\frac{100}{2^3}$ $6 \times 6 < 100$ 6, 5, 4, . . 3, 2, 1

5 → [7 . . . . . 11] ≤ 6 9 $\frac{100}{2^4}$ $9 \times 9 < 100$ 9, 8, 7

6 → [10, 11] ≤ 3 ↑ 10 $10 \times 10 == 100$

[1]

≤

#steps : 6 steps.

$100 \rightarrow \frac{100}{2} \rightarrow \frac{100}{2 \times 2} \rightarrow \frac{100}{2 \times 2 \times 2} \rightarrow \ldots\ldots 1$

steps

$\log_2^N \Rightarrow \log_2^{100}$

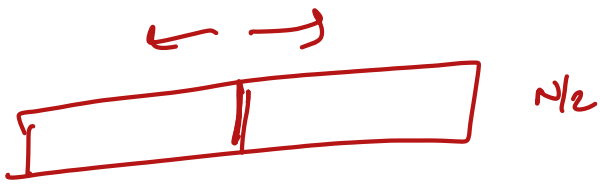
$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \rightarrow \cdots$

steps

$\log N$

iterations = $\log_2 N$

Binary Search



iterations: $\log_2 N$

1 iteration $\Rightarrow$ 1 ns

Approach 1 $(\sqrt{N})$

$N = 2^{32}$

2^{16} ns

$= 10{,}000$ years

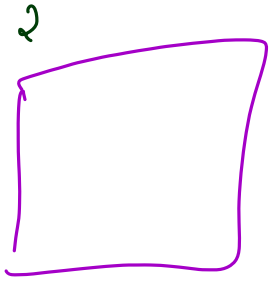
Approach 2 $(\log_2 N)$

$\log_2 2^{32} = 32$ ns

$\log_b b^a = a$

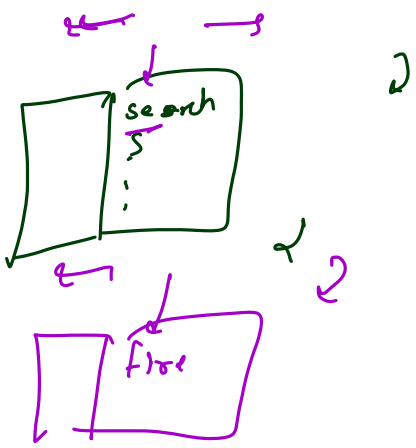
Data Structures

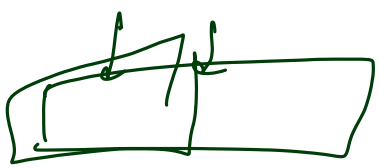
→ Data Structures are a means of organizing and storing data so that we can perform operations efficiently.

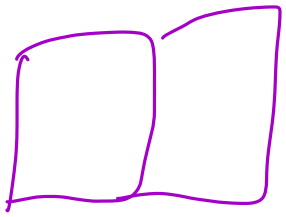


Dictionary

door







Novel

door

Hashmap

Glossary / Appendix

door : [11, 13, 156, ...]

fan : [1, 5, 11, 100..]

door

Trees

Undo / Redo : stack

Auto Complete : Trie

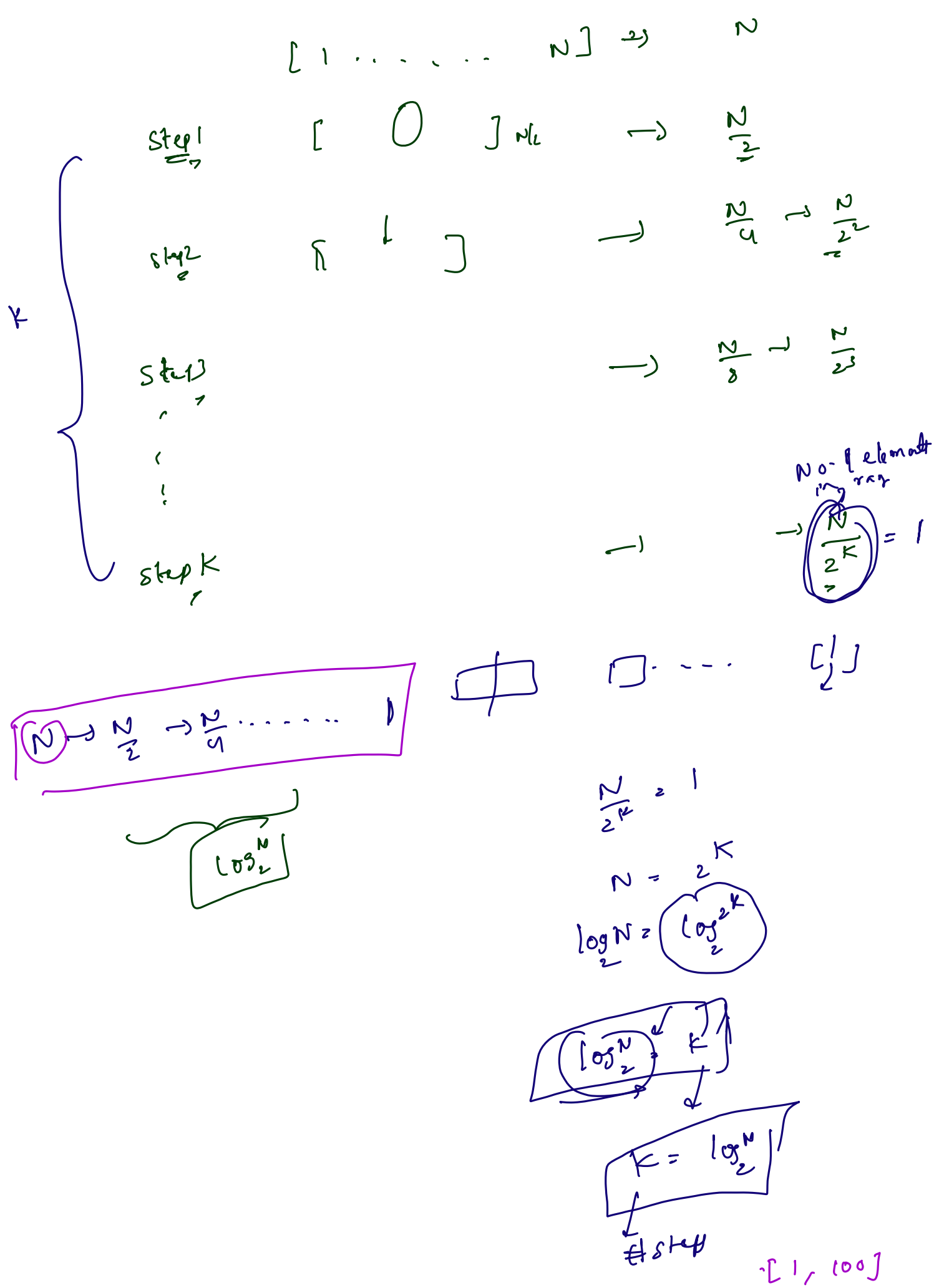
Google Maps / Facebook : Graphs

Data Structures / Algorithms

→ Yahnit Sirineni

→ Slack

→ yahnit.sirineni_1 @ scaler.com


[1 . . . . . N] → N
Step 1
[0] N/2
→ N/2
step 2
[1]
→ N/4 → N/2^2
K
step 3
→ N/8 → N/2^3
step K
→ 1
No. of elements in range
N/2^K = 1
[1]
N → N/2 → N/4 1
log2 N
N/2^K = 1
N = 2^K
log2 N = log2 2^K
log2 N = K
K = log2 N
# step
[1, 100]
mid = 50

sqrt(N)

low = 1, high = N;

50*50 > 100

[1, 49]

high = mid-1

while (low ≤ high) {

mid = (low + high) / 2;

if(mid * mid > N) {

high = mid - 1;

}

else if(mid * mid < N) {

low = mid + 1;

}

else if(mid * mid == N) {

return mid;

}

}

B.S

[1,

mid: 8

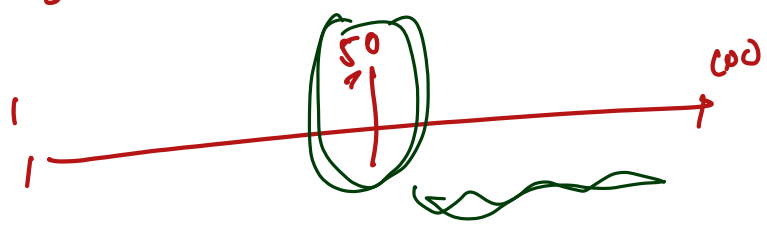
8*8 < 100

↳ 9

9 6 5

low

Search: [1, N]



⇒ B.S

→ Maximum / Minimum

→ N = ←

Time Complexity

C

LS

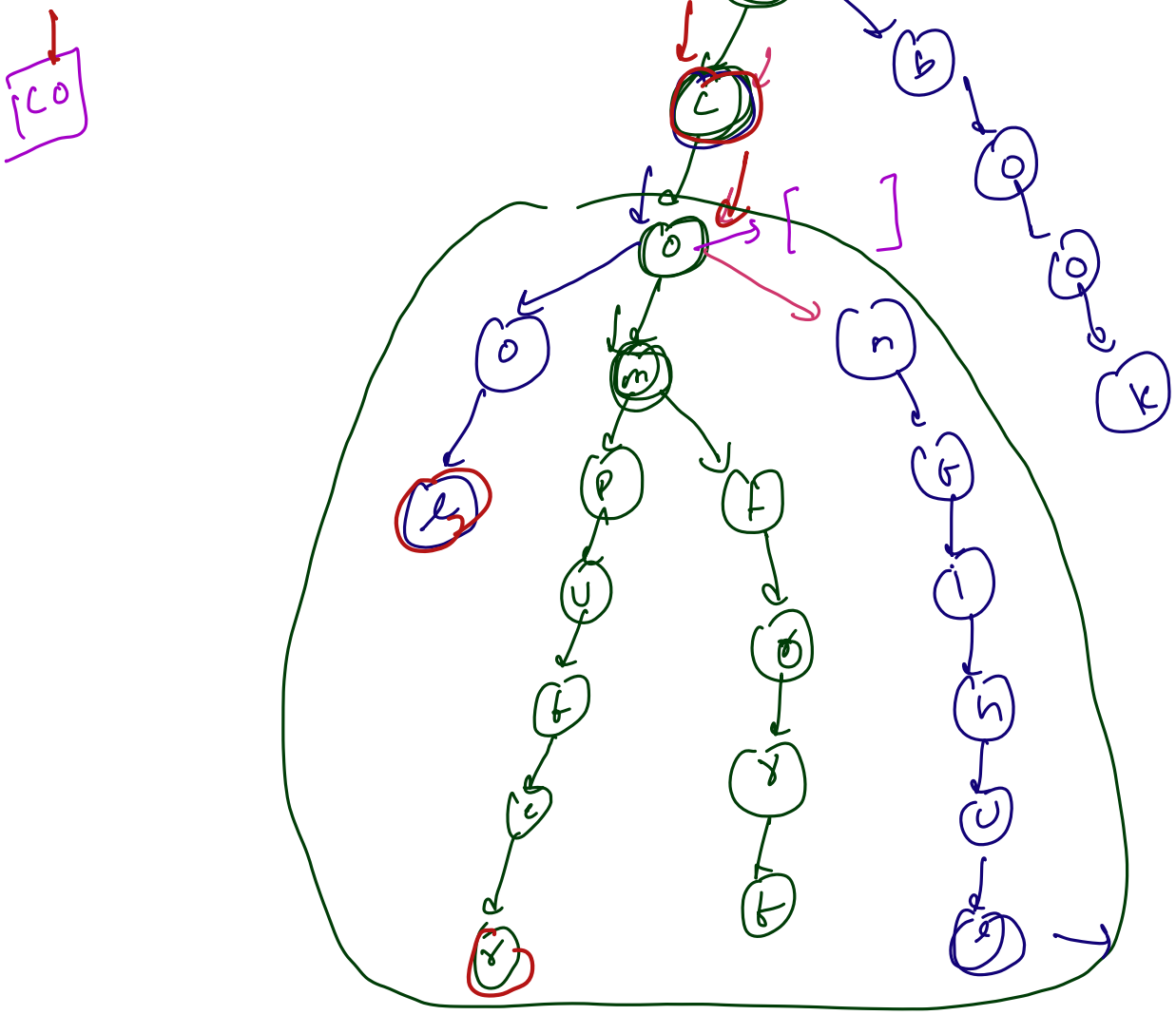
lowercase

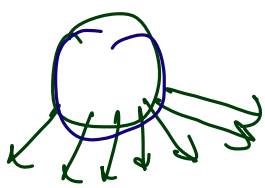
S = [computer, comforter, continue, cool]

bool

Trey

26 children





Address of 26 children

Input 2

Input → Algo → Output

bug