Introduction to Problem Solving

Lecture will begin shortly…

Vinay Neekhra

Senior Instructor & Mentor

Say hi 👋🏻 in Scaler Lounge

"Look around you. Every man made object around you is a solution to a problem humanity faced!"

Q. The person is in Banglore and wants to be in Delhi.

→ try to understant the content

Question → question tab / public chat
to: everyone
ans : to: me
private chat

① → time → as soon as possible

soln ⇒ ? flight → Private jet

② → money ⇒ train

→ helicopter

the person is prime minister

→ 20,000 years ⇒ walking

→ Business

10:39

Pillars of problem solving

① Understanding the problem

② observations

③ Coming up with multiple soln

④ Quality assurance (dry run)

③ → money & time (as little as possible money)
as fast as possible

FAQ.
① notes will be uploaded after the class
→ 1.5x, 2x speed, bookmarks.
② Hw/ assignments would be unlocked after the class.
③ There is no deadline for assignments (Hw.
④ 2 Hours 30 min. than doubt session.

Peer learning: Network

→ Slack, whatsapp DM

→ Office Hour. → Thursday 9pm – 11pm

- one on one ~~discussion~~
- → Study Groups
- → General chit-chat

Link:

→ Slack, whatsapp

Q. Count of factors: 10 is divisible by 2?

↓

any no. which divides N completely

ex 1

$N \% i == 0$

$N = 24 \Rightarrow \{1, 2, 3, 4, 6, 8, 12, 24\} \Rightarrow$ 8 factors.

$N = 10 \Rightarrow \{1, 2, 5, 10\} \Rightarrow$ 4 factors.

Brute force: → easiest to implement soln

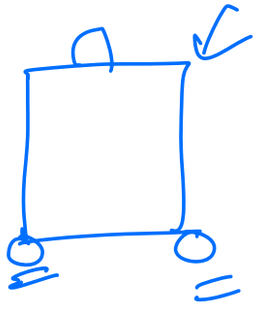
→ explore all the possibilities.

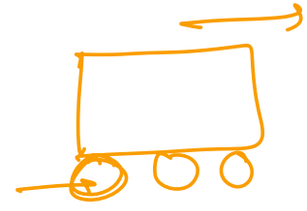
walk, horses, flight, portkey, broom.

→ wheel ⇒ ? 5000 BCE

→ trunk ⇒ < 5000 years back

<5000 BC

⇒  ⇒ 1970's



pseudo code:

```
int CountOfFactors(int N) {
    int factors = 0
    for (int i=1; i<=N; i++) {
        // i is a factor?
        if (N % i == 0)
            factors += 1
    }
    return factors.
}
```

10 → (1 → 10) N times

N → N iterations

→ 10^8 iterations per sec.

1 GHz ⇒ 10^9 pulse/sec

1 instⁿ ⇒ 10 pulse

N	iterations
10^8	10^8
10^9	10^9
→ 10^{18}	10^{18}

execution time

1 sec.

10 secs.

10^{10} sec. = 317 years.

10×10^9 sec

10 Billion sec.

↓

1 million sec → 12 day

1 billion → 31.7 years

10^8 → 1 sec

1 inst → $\frac{1}{10^8}$ sec

10^9 instⁿ → $\frac{10^9}{10^8}$ sec

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

unitary method

→ Solⁿ ✓

practical solⁿ ✗

$2^5 = 32$

⇒ Obsⁿ.

i is a factor ⇒ N % i == 0

⇒ J = N / i ⇒ ? factor

$\Rightarrow$ $i * J = N \Rightarrow$ (i, J are factors)

N = 24

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

N/2 ?

N = 25

i		N/i
1	<	25
5	<=	5
25	>	1

N = 36

i		N/i
1	<	36
2	<	18
3	<	12
4	<	9
6	<	6

i	N/i
9	> 4
12	> 3
18	> 2
36	> 1

Obs^n ① After a certain number, factors are repeating

② inflection point cond^n

$$i <= N/i$$

$\Rightarrow \quad i <= N/i$

$\Rightarrow \quad i*i <= N$

$\Rightarrow \quad i <= \sqrt{N}$

optimised code

```
1. int countFactors (int N) {
2.     int factors = 0
3.     for (i = 1; i*i <= N; i++) {
4.         if (N % i == 0) {
5.             if (i == N/i)
6.                 factors += 1
7.             else
8.                 factors += 2
           }
3      }
```

$N = 25$ $\quad 6 <= 5$

i → 1 2 3 4 5 6

factors → 0 2 4

factors ⇒ 4

$25 \Rightarrow \{1, 5, 25\}$

$N = 24$ $\quad 2 <= 4$

i → 1 2 3 4 5

f → 0 2 4 6 8

(8)

$\sqrt{N}$ iteration

9. return factors.

10^8 iteration / sec.

execution time.

N	iterations	time
10^{18}	10^9	→ 10 sec

317 year ↓

Q. Given N, check if it's prime or not. $N > 0$

N = 10 ⇒ No ⇒ (1, 2, 5, 10)

Quiz ✓ ✓ ✓ ✗ ✗ ✓

10, 11, 23, 2, 25, 27, 31 ⇒ 4

Prime: number which is having exactly 2 factors. (1 and itself)

```
boolean checkPrime(int N){
    if (countOfFactors(N) == 2)
        return true;
    else
        return false
```

$\sqrt{N}$

}

Abstraction: ignoring lower level details

→ N ⇒ 1. ⇒ neither is prime or composite

↓

2 factors.

_____ × _____

Q 4th class.

1 ⟶ 100 →

⇒ ↓ 15 secs. Gauss

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

$$+$$

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

$\Rightarrow$ 101 101 ... 101 101 101

25 $\Rightarrow$ $(101)^* 100$

$\Rightarrow$ $S = \underbrace{(101)^* 100}_{2} \Rightarrow (101)^* 50 \Rightarrow \underline{5050}$

$\Rightarrow$ Sum of N natural numbers.

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

$S = N + N-1 + \ldots + 2 + 1$

$2S = (N+1)^* N$

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

assume valid inputs

Q Given a no. N → perfect square. find sqrt(N).

$N = 25 \Rightarrow ans = 5$ $N = \{1 \longrightarrow 25\}$

⇒ N = 100 ⇒ ans = 10

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

$\sqrt{N}$ ⇒ $i*i \le N$ ⇒ $i <= N/i$

N iteration → $\sqrt{N}$

N − 1 to 2 → 2 time

1 to 5 → 5 times

Amazon MCQ

(a) $\log_2 N$

(b) N

(C) $\sqrt{N}$ ✓

(D) None of these

Q. Find sqrt N.

If N is not a perfect square, return floor(sqrt(N)).

N = 49 ⇒ 7

N = 60 ⇒ 7

N = 31 ⇒ 5

$1*1 <= 50$

$\sqrt{31}$ ⇒ LS. ⇒ 5

N = 29 ⇒ 5

N = 50

i	i * i <= N
1	~~ans = 1~~
2	ans = 2
3	ans = 3
4	ans = 4
5	ans = 5
6	~~ans = 6~~
7	ans = 7
8	64 <= 50 X

N = 29 ⇒ 5 (ans)

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

ans = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

return 5

floor(x) ⇒ greatest integer <= x

floor(7.258) = 7

floor(6.99) = 6

floor(9) = 9

```
1. int sqrt(N){
2.     int i=1, ans=0
3.     while(i*i<=N){
4.         ans=i
5.         i+=1
       }
6.     return ans;
}
```

iterations ⇒ √N

i → {1 to √N}

Binary search $\log N$

⊛ log Basics: ↓

$\log_a b = c$,

$\Rightarrow a^? = b$

$\log_2 32 = 5$

$\Rightarrow 2^5 = 32$

① $\log_2 4 = ?$ (x) 2

② $\log_2 64 = ?$ (6)

③ $\log_5 25 = 2$

④ $\log_2 (2^6) = 6$

$\Rightarrow \log_2 2^n = n$

$2^x = 4 \Rightarrow$

$2^x = 64 \Rightarrow$

2

$1 = 2^0$

$2 = 2^1$

$4 = 2^2$

$8 = 2^3$

$16 = 2^4$

$32 = 2^5$

$64 = 2^6$

(5) $\log_3 81 = 4 \Rightarrow 3^4 = 81$

$\Rightarrow 2^k = N$

$\Rightarrow \log_2 N = k$

Q. Given an integer N (+ve). How many times do we need to divide it by 2 until it reaches 1.

$N = 100$
↓/2
50
↓/2
25
↓/2
12
↓/2
6
↓/2
2

$N = 324$ [Homework]
↓/2
162
↓/2
81
↓/2
40
↓/2
20
↓/2

3
↓/2
1

ans = 6

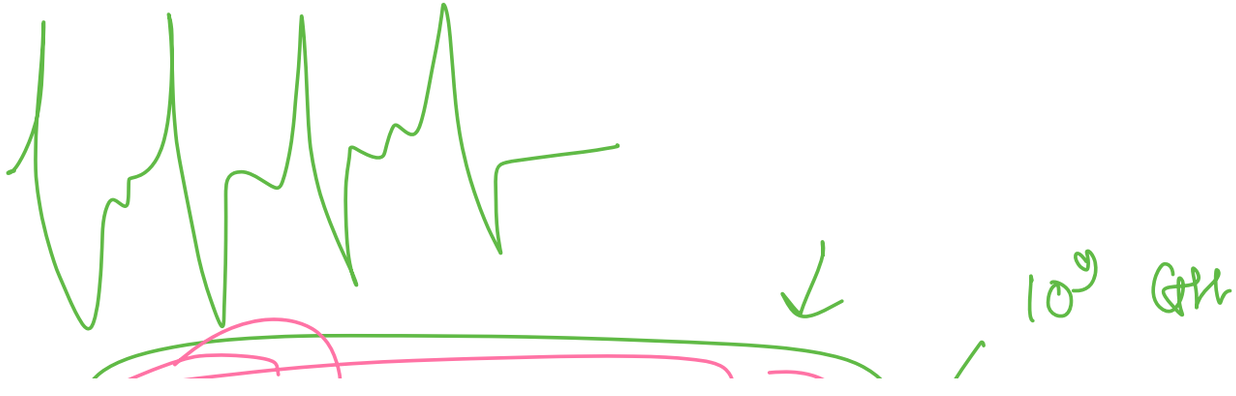
10
↓/2
5
↓/2
2
↓/1
1

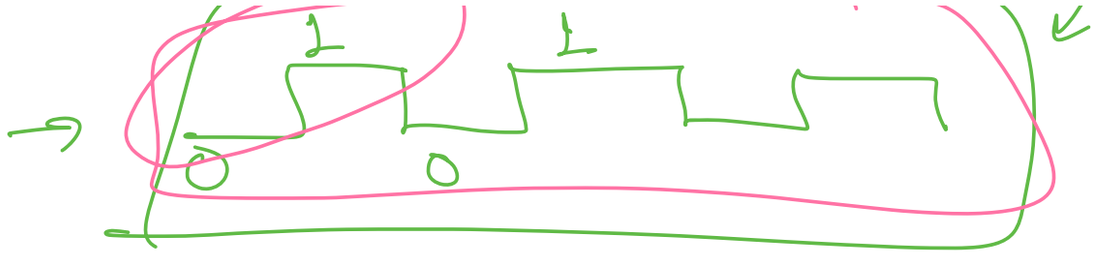
ans = 8

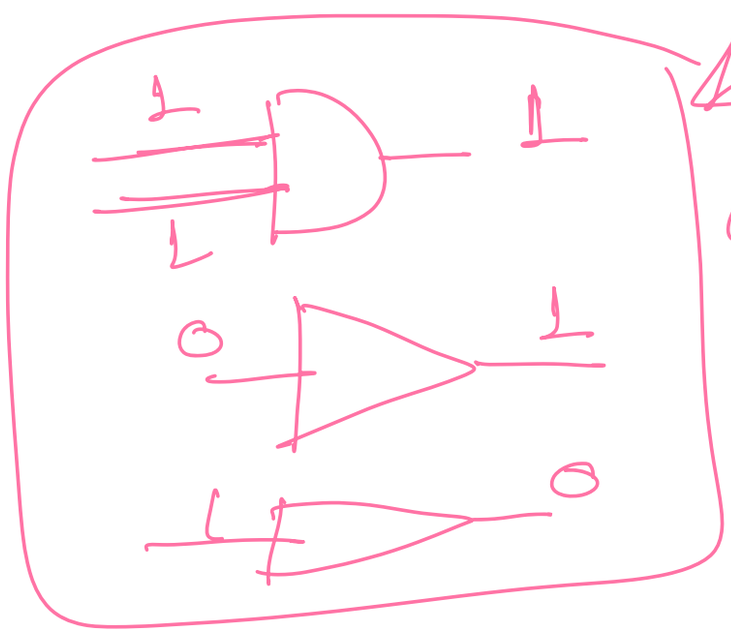
⇒ walking → Horses → . . .

↓

Rockets → Broom (X)







logic gates

AND
NAND
OR
NOR

4 pulses

1 0 0 1

5 → 1 0 0 1
+ 4

C = a + b ≈ 10 Per