Q1. Given a no. check if its a prime or not.

ebay

N

1 ⟶ neither prime nor composite.

19 ⟶ prime

21 ⟶ not prime

63 ⟶ not prime

79 ⟶ prime

Prime No. : Any positive no. that has exactly 2 factors.

Brute force

* Check all possible factors of a no. N,
  if count of factors == 2, then return true
  else return false.

ex => 14

1 2 3 4 5 6 7 8 9 10 11 12 13 14

✓ ✓ ✗ ✗ ✗ ✗ ✓ ✗ ✗ ✗ ✗ ✗ ✗ ✓

c = 4, => not prime.

17

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

✓ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗ ✓

C = 2 ⇒ prime.

code

```
boolean checkPrime(N) {
    C = 0
    for(i = 1; i <= N; i++) {
        if(N % i == 0) {
            C++
        }
    }
    if(C == 2)
        return true
    else
        return false.
```

for i/p ⇒ N, total iterations = N.

assumptions

10^8 iterations ⟶ 1s.

N = 10^9; iterations = 10^9.

10^8 i ⟶ 1s

⇒ 1 i ⟶ $\frac{1}{10^8}$ s

⇒ 10^9 i ⟶ $\frac{1}{10^8} \times 10^9 = \underline{\underline{10s}}$

5 choc. ⟶ 10 Rs.

1 choc ⇒ $\frac{10}{5}$

15 ⇒ $\frac{10}{5} \times 15$.

$N = 10^{18}$

10^8 i $\longrightarrow$ 1s

$\Rightarrow$ 1 i $\longrightarrow \frac{1}{10^8}$ s

$\Rightarrow$ 10^{18} i $\longrightarrow \frac{1}{10^8} * 10^{18} = 10^{10}$ s.

$\simeq$ 317 yrs.

checkPrime (10^{18}) $\longrightarrow$ 317 yrs.

Optimisations:

if. a, b, N are three integers (+ve)

& if. $a \times b = N$, $\Rightarrow b = N/a$.

a, b are factors of N.

$\Rightarrow$ a, N/a are factors of N.

if. a is a factor of N, then N/a is also a factor.

$\Rightarrow$ eg: N = 24.

1 2 3 4 6 8 12 24 |

i	N/i
1	24.
2	12
3	8
4	6
6	4
8	3
12	2
24	1

N = 100.

$i \leq N/i$

$\Rightarrow (i = N/i)$ max^m for i

$\Rightarrow (i^2 = N)$ max^m

$\Rightarrow \boxed{i_{max} = \sqrt{N}}$

i	N/i
1	100
2	50
4	25
5	20
10	10
20	5
25	4
50	2
100	1

<= >=
$a <= 57$
$a_{max} = 57$
$a <= x$
$\Rightarrow a_{max} = x$

Code

```
boolean checkPrime(N){
    c=0;
    for(i=1, i*i<=N; i++){
        if(N%i==0)
            if(i==N/i)
                c++
            else
                c=c+2
    }
    if(c==2)
        return true
    else
        return false.
```

dry run

N = 27

i	c
1	2
2	2 (x)
3	4
4	4 (x)
5	4 (x)

i/p ⇒ N, no. of iterations = $\sqrt{N}$ $[1 - \sqrt{N}]$.

10^8 i → 1s.

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

10^8 i → 1s

⇒ 1 i → $\frac{1}{10^8}$ s

⇒ 10^9 i → $\frac{10^9}{10^8} = \underline{10s}$

power of observation

$4^{\underline{m}}$ ⇒ 1 + 2 + 3 + 4 + ——— + 100.

Carl Friedrich Gauss

S = 1 + 2 + 3 + 4 + ——— + 97 + 98 + 99 + 100.

S = 100 + 99 + 98 + 97 + ——— + 4 + 3 + 2 + 1.

2S = 101 + 101 + 101 + 101 + ——— + 101 + 101 + 101 + 101

⇒ 2S = 100 * 101

$$\Rightarrow \boxed{S = \frac{100 * 101}{2}}$$

For N nos.

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

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

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

$$\Rightarrow 2S = N * (N+1)$$

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

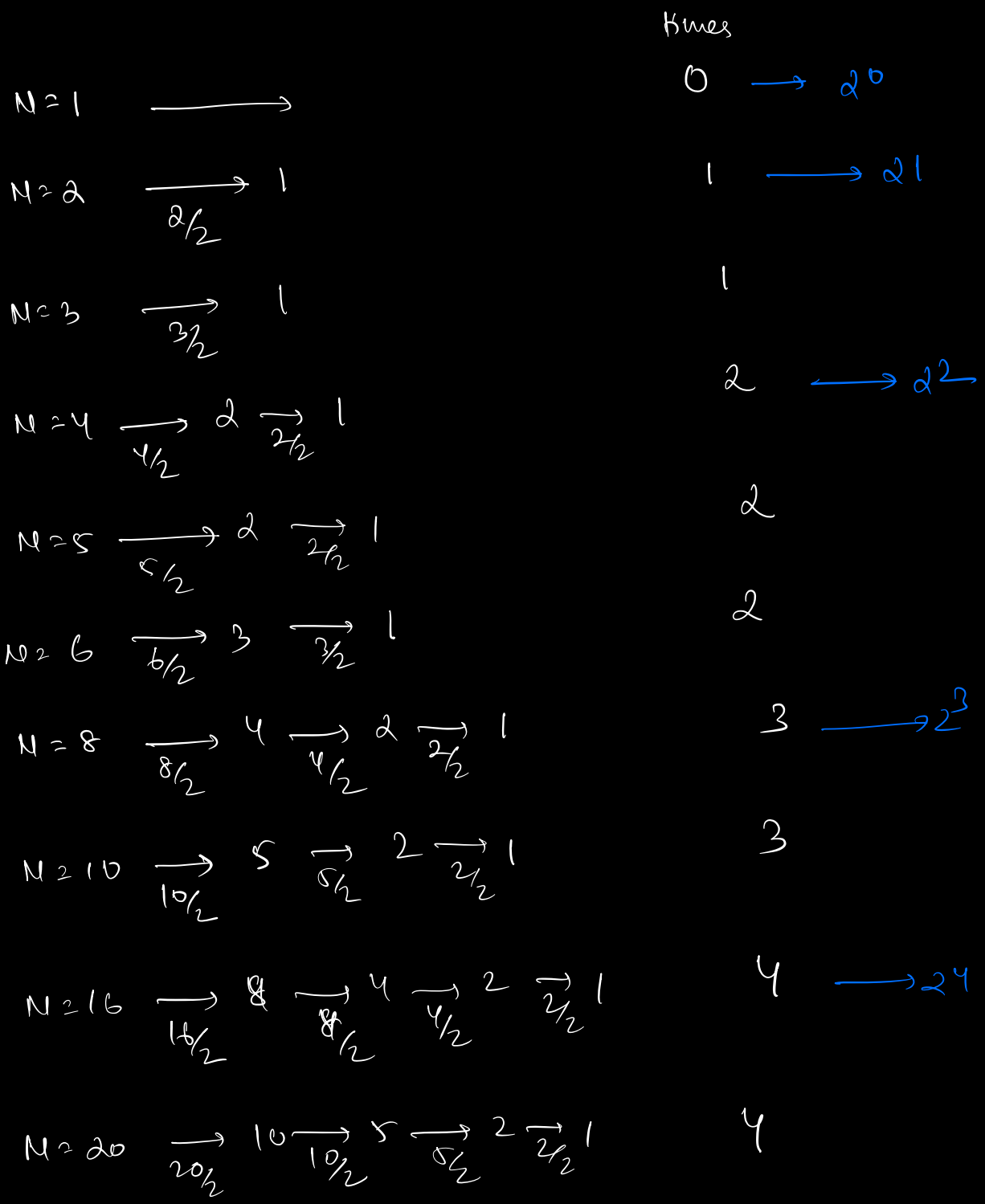
S = Sum of first N natural nos.

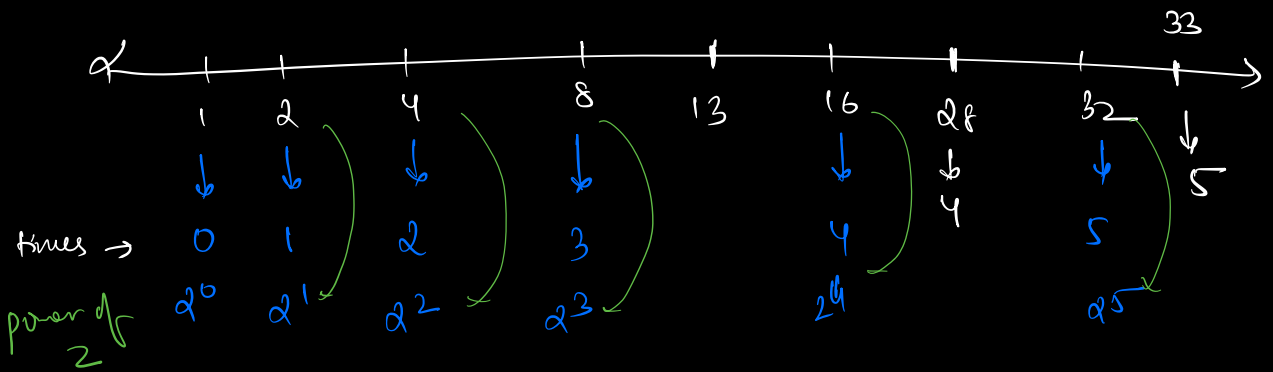
Q. Given N, how many times we need to divide it by 2 till it becomes 1.

division → integer division

$N = 7 \xrightarrow[7/2]{} 3 \xrightarrow[3/2]{} 1$ ∴ times = 2

$N = 15 \xrightarrow[15/2]{} 7 \xrightarrow[7/2]{} 3 \xrightarrow[3/2]{} 1$: times = 3.


Knes
N=1
N=2
N=3
N=4
N=5
N=6
N=8
N=10
N=16
N=20
0 → 2^0
1 → 2^1
2 → 2^2
3 → 2^3
4 → 2^4



$33 \xrightarrow[33/2]{} 16 \xrightarrow[16/2]{} 8 \xrightarrow[8/2]{} 4 \xrightarrow[4/2]{} 2 \xrightarrow[2/2]{} 1$ ⇒ times = 5

$28 \xrightarrow[28/2]{} 14 \xrightarrow[14/2]{} 7 \xrightarrow[7/2]{} 3 \xrightarrow[3/2]{} 1$ ⇒ times = 4

$13 \xrightarrow[13/2]{} 6 \xrightarrow[6/2]{} 3 \xrightarrow[3/2]{} 1$ times = 3.

$N \xrightarrow[N/2^1]{} N/2 \longrightarrow N/4 \xrightarrow[N/2^2]{} N/8 \xrightarrow[N/2^3]{} \cancel{N/16} \xrightarrow[N/2^4]{} \cdots = 1 = \frac{N}{2^k}$

$$\frac{N}{2^k} = 1$$

$$\Rightarrow \quad N = 2^k \quad \Rightarrow \quad \log_2 N = \log_2 2^k$$

$$\Rightarrow \boxed{k = \log_2 N}$$

$\log_a a^k = k$

$\log_2 2^k = k$

$\log_2 25 = 4$ ⟷ $qp = 4$ (dono)

Amazon

Q Given a perfect sq., find the sq. root of the no.

perfect sq:- N is a perfect sq. if $\sqrt{N}$ is an integer, then exists an integer x, such that

$x \times x = N$.

$N = 25 \rightarrow 5 \times 5$ ✓

$N = 100 \rightarrow 10 \times 10$ ✓

$N = 49 \rightarrow 7 \times 7$ ✓

$N = 28 \rightarrow$ ~ -- (X).

Check for all nos. from 1 to N, to find sq. root

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

N	iterate
49	7
25	5

Total no. of iterations = N (X).

↓

Since, N is always a perfect sq. max iterations = $\sqrt{N}$.

time calculation

$N = 2^{64}$, iterations = $\sqrt{N} = \sqrt{2^{64}} = 2^{32}$.

10^8 i → 1s

$2^{32} \Rightarrow \frac{1}{10^8} \times 2^{32}$

$= \frac{2^{10} \times 2^{10} \times 2^{10} \times 2^2}{10^8}$

$2^{10} = 1024 \approx 10^3$

$$= \frac{10^3 \times 10^3 \times 10^3 \times 4}{10^8}$$

$$= \frac{10^9 \times 4}{10^8} = 40s.$$

$N = 2^{64} \Rightarrow$ Time $= 40s$

Obs

ans $\Rightarrow$ $[1, N]$ $\rightarrow$ ans always lies from 1 to N

$N = 100$

$\frac{N}{100} \Rightarrow$ [1 2 3 4 ---- 49 50 51 — — 98 99 100]

$\times 2$

$50 \times 50 = 100$

$50 \times 50 = 2500 > 100.$

$\frac{N}{50} \Rightarrow$ [1 2 3 4 — 24 25 26 — 47 48 49]

$\times 2$

$25 \times 25 = 100$

$25 \times 25 = 625 > 100.$

$\frac{N}{25}$ $\times 4$ $\Rightarrow$ [1 2 3 4 .. 10 11 12 — 23 24]

$12 \times 12 = 100$

$12 \times 12 = 144 > 100.$

$x/8 \Rightarrow$ [~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6 7 8 9 10 11]

6 × 6 ~~≥ 100~~

6 × 6 ≱ 36 < 100

$x/16 \Rightarrow$ [~~7~~ ~~8~~ 9 10 11]

9 × 9 ~~≥ 100~~

9 × 9 ≱ 100

$x/32 \Rightarrow$ [10 11]

10 × 10 = 100

$N \longrightarrow N/2 \rightarrow N/4 \rightarrow N/8 \quad \cdots\cdots \quad 1$

no. of steps = $\log_2 N$

i/p N → iterations = $\log_2 N$.

$N = 2^{64} \Rightarrow i = \log_2 2^{64} = 64$

Time = $\frac{64}{10^8 s} = 0.00000064 s$

Binary Search → [2 dedicated class]

upcoming

1) Time & Space comple. (2)
2) Arrays → 6
   - Intro to arrays
   - Prefix sum
   - Carry forward
   - Subarrays | Sliding window | Combib
   - 2D matrix
   - Interview problems
3) Bit manipulation → (2/3)
4) Maths & Arrays → (2)
5) Sorting | Strings | Hashmap → 4

6) Recursion → 2

7) Subsets & subsequences → 2

8) Linkedlist → basics (1)

9) Stacks → basics (1)

10) Binary tree → basics (2)

Doubts

```
boolean checkPrime(N){
    c=0;
    for(i=1, i*i<=N; i++){
        if(N%i==0)
            if(i==N/i)
                c++
            else
                c=c+2
    }
    if(c==2)
        return true
    else
        return false.
```

i	c	i*i
1	2	1
2	2	4
3	4	9
4	4	16
5	4	25
6	4	36
7	4	49
8	4	64
9	6	81
10	6	100

$$50 \xrightarrow{/2} 25 \xrightarrow{/2} 12 \xrightarrow{/2} 6 \xrightarrow{/2} 3 \xrightarrow{/2} 1$$

$$\frac{50}{2\times2\times2\times2\times2} = \frac{50}{2^5}$$

ans = 5 times to reach 1.

$$\frac{N}{2} \to \frac{N}{4} \to \frac{N}{8} \to \frac{N}{16} \cdots \frac{N}{2^k}$$

$$\frac{N}{2^k} = 1$$

$$\Rightarrow N = 2^k$$

$$\Rightarrow \log_2 N = \log_2 2^k$$

$$\Rightarrow \boxed{\log_2 N = k}$$

$\log_2 N$

$$\log_2 2^5 = 5 \log_2 2$$

$\log$

$=$

$$25 \xrightarrow[25/2]{} 12 \cdot 5 \xrightarrow[12.5/2]{} 6 \cdot 25$$

$$\log_2 25 \Rightarrow \log_2 2^4 \rightarrow \log 16$$

$$= 4$$