PRIME Numbers

Prime → exactly 2 divisors → 1 & N

{ 7: 1, 7   13: 1, 13

not prime { 8: 1, 2, 4, 8   9: 1, 3, 9

Prime Check

O(N)

1 ——— N Count div

I)
```
cnt = 0
for (i=1 —— N) {
    if (N%i == 0) {
        cnt++;
}
```

II)
```
if (cnt == 2) {
    Prime
```

3

$(\cancel{1}, 2, 3, \ldots \text{N-1}\ \cancel{N})$

as soon as you atleast 1 div in the range (2, N-1)

break

Algo-2

O(N) in worst case

N = 7

7 % 2	NO
7 % 3	NO
7 % 4	NO
7 % 5	NO
7 % 6	NO
7	i=7

```
for(i=2, i<=N-1 , i++){
    if( N%i==0){
        break;
    }
}
if(i==n){
    prime
}
else{
    Not PRIME
}
```

i= 3

3

loop ends
i==n

break the loop
i<=n-1

2, 3 ---- 8
X 9

i==7
2, 3, 4, 5, 6 7
i++

2 3, ------ 8 9

N = 9

2	9 % 2	No	
3	9 % 3	Yes	Stop

faster for non-prime No's

8

Algo-3   Root N optimisation

$2 \quad \sqrt{N} \quad N-1$

Examples -

$N = 30$ $\cancel{1}, 2, 3, 5, 6, 10, 15, \cancel{30}$

$\cancel{1 \times 30 = 30}$

inc

$2 \times 15 = 30$

$3 \times 10 = 30$

$5 \times 6 = 30$

dec

Small   Big

$\leq \sqrt{N}$

Not Prime → any 1 divisor

Yes

$\sqrt{30} = 5.5XX$

$a == b$

$a^2 = N$

$\Rightarrow a = \boxed{\sqrt{N}}$

$\boxed{a} \cdot \boxed{b} = N$

$> \sqrt{N} \quad > \sqrt{N}$

$= > N$

$a \times b = N$

$\Rightarrow$

$1 \times b = N$

$2 \times b = N$

$\vdots$

$\sqrt{N}$

$36 = 1, 2, 3, 4, 6, 9, 12, 18, 36$

$\leq \sqrt{N}$

$1 \times 36 = 36$

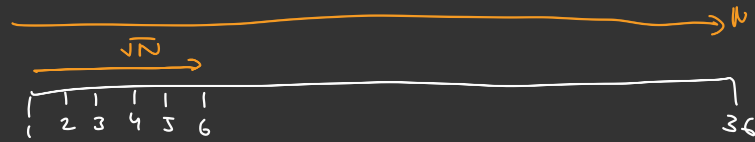
$2 \times 18 = 36$

$3 \times 12 = 36$

$4 \times 9 = 36$

$6 \times 6 = 36$

Pairs



$N = 10^4$

$\sqrt{N} = 100$ Steps

$N = 10^6$

$\sqrt{N} = 1000$ Steps

```
isPrime(n) {

    isPrime = true;

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

        if (N % i == 0) {
            isPrime = false;
            break;

    if (isPrime)  -> "Prime"
    else          -> not Prime
```

3

3

$i * i \leq N$

$\Rightarrow i^2 \leq N$

$O(\sqrt{N})$

$O(1)$

}

$N = 5$	$i=2$	$2^2 \le 5$	$5 \% 2$	NO	
	$i=3$	$3^2 \le 5$	out of the loop		Prime
$N=3$	$i=2$	$2^2 \le 3$	out of the loop		Prime
$N=36$	$i=2$	$2^2 \le 36$	$36 \% 2$	Yes	Not Prime
$N=29$	$i=2$	$2^2 = 4 \le 29$	$29 \% 2$		
	$i=3$	$9 \le 29$	$29 \% 3$		
	$i=4$	$16 \le 29$	$29 \% 4$		
$\sqrt{N}$	$i=5$	$25 \le 29$	$29 \% 5$		
Steps	$i=6$	$36 \le 36$	out		[Prime]

Problems involving $\Rightarrow$ Range of Prime Numbers

Q Generate all Primes in Range 1 to N.

N = 100 2, 3, 5, 7, 11, - - - - - - - - 97

Brute force

O ($N\sqrt{N}$)

```
for(i=1, i<=N; i++){
    if(isPrime(i)){ print(i) };
}
```

$\sqrt{N}$

$2 \longrightarrow \sqrt{N}$

Prime Sieve - Sieve of Eratosthenes.

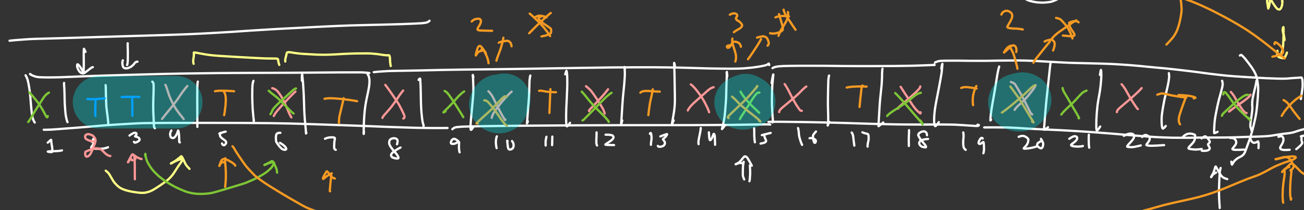
(famous technique)

Init ⇒ "all Prime."

{2, 3, 7, 11, 13, 17, 19, 23}

Prime Yes

boolean primes



2 → 4

3 → 6, 9, 12

4 → 8, 12, 16

i = 2

i = 3

i = 4

for (i = 2, i <= √N ; i++) {

if (primes[i] == true) {

for prime no's →

for (j = 2i ; j <= N, j = j + i) {

primes[j] = false,

}

}

}

Mark all multiples of 2 as Not Prime

4, 6, 8, 10, --

j = ↑ ↑ ↑

3 → 15 3 → 6, 9, 12, 15, --

5 → 15 5 → 10, 15, 20, 25 --

optimise it

$j = i^2$

Print

for (i = 2 —— N) {

if (isPrime(i)) print i

}

2 —— N

3, 6, 9, 12, 15, 18, --

Work = $\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \text{-----} \frac{N}{P}$ + √N

= O(N log log N)

largest prime ≤ √N

$i = 2$

$i = 3 \rightarrow$

$i = 4$

$i = \boxed{5} \rightarrow$

6, 9, 12, 15, 18, 21, 24

10, 15, 20, 24, $\boxed{25}$

$j = 10$, 15, 20

first multiple i will work

$\approx O(N)$ → very small

Practically

$j = \cancel{2i}$; $j <= n$; $j = j + 1$

i^2

$\boxed{N}$

$2 \text{ ---- } \sqrt{25}$

$= 2 \text{ ---- } 5$

$2 \text{ ---- } \sqrt{24}$

$= 2, 3, 4$

```
primes[N+1] = true,

for (i=2,  i<=√N ,  i++) {

    if ( isPrime[i] ) {
            for (j= i^2 ; j<=N ; j= j+i ) {
                isPrime[j] =  false;
    }       }

}

for(i=2 ________ N){
    if (isPrime(i)) {   print(i) };

}
```

Q) given Q Queries, of the form [a,b] find out the count of primes in range [a,b]. $a, b \le 10^6$

Q = 5

a b

[3, 10]

[10, 100]

[5, 20]

[2000, 10000]

[1500, 3500]

optimised Algo →

1) Precompute sieve + prefix array $\quad N \log \log N + N$

2) iterate over ~~3 to~~ 10 and count ✓

(b - a)

for each query

prefix[i] = count of primes till i

$Q(a,b) = prefix[b] - prefix(a-1)$ → O(1) per query

Overall

$N \log \log N + Q$ done

10.25

Prime Factorisation

↳ No = Product of Powers of Primes.

$48 = 2^4 \cdot 3^1$

$51 = 3 \cdot 17$

$20 = 2, 2, 5$

2	48
2	24
2	12
2	6
3	3
	1

Algorithm-1

```
for (i=2; i<=n; i++) {
    if (n % i == 0) {
        cnt = 0
        while (n % i == 0) {
            n = n / i
            cnt = cnt + 1;
        }
        print (i, cnt);
    }
}
```

$2^1 \cdot 3^3$

2	54
3	27
3	9
3	3
	1

N

(2,1) (3, 3)

$\boxed{N}$

i = 2   27   54   yes

54 ↓ 27   cnt=1

yes   cnt=0

9   cnt=1

3   cnt=2

1   cnt=3

i=3   1

i=4 $\boxed{1}$

worst case

no is prime → $\boxed{O(N)}$

$n=11$

2 ✗
3 ✗
4 ✗
5 ✗
6 ✗
7 ✗
8 ✗
9 ✗
10 ✗
$\boxed{11 ✓}$

$\boxed{\text{4 steps.}}$

$\frac{11}{11} = 1$

$n = 72$

$i = 2$ 36 $cnt = 1$

18   $cnt = 2$

9   $cnt = 3$

$2^3 \; 3^2$

$= 8 \times 9$

$= 72$

$i = 3$ $n = 9$

3   $cnt = 1$

1   $cnt = 2$

$i = 4$ $n = 1$ Stop

Algo-2

optimised

TC

$O(\sqrt{N})$

Not True $\sqrt{N}$

WHY ?

$4 \le \sqrt{11}$

```
for (i=2; i<=n; i++) {

    if (n%i == 0){
        cnt = 0
        while(n%i == 0){

            n = n/i
            cnt = cnt + 1;
        }
        print (i, cnt);

    }
}
if(n>1) { print(n, 1); }
```

outside for loop

$\sqrt{11}$ Prime 1

$< \sqrt{N}$ — N

$N = 44$

$2^2 \cdot 11 = 4 \times 11 = 44$

i	n	cnt
2	44	0
2	22	1
2	11	2
3	11	–
4	11	–
5	11	–
6	11	–
7	11	–
8	11	–
9	11	–
10	11	–

Stop $\sqrt{N}$

must be prime no as we couldn't find any more or $\sqrt{N}$

$11 \le 11$

1

$cnt = 0$

$cnt = 1$

$N = 10$

	n	cnt
2	10	0
2	5	1

$\sqrt{5}$

$\approx$

2.2

$3 < \sqrt{5} \longrightarrow$ Stop

2×5

```
if (n > 1) {
    print(n)
}
```

2	16	
	8	1
	4	2
	2	3
	$N == \boxed{1}$	4

$\to (2, 4)$

$5^2 \le 55 \Rightarrow 25 \le 55$

6^2 36 ≤ 11

$\boxed{6^2 \le 11}$

for (i=2, i^2 <= 11 ; i++){

Stop

3

if (n>1){

print(11,1);

3

N = 110

i=2	110	cnt=0	
	55	cnt=1	(2', 5', 11')
i=3	55	x	= 2x5x11
i=4	55	x	= 110
i = 5	55	cnt=0	
~~i=6~~	11	cnt=1	
i=6	→ 11	x	NO

Prime NO

$\sqrt{11} = 3$

$\sqrt{N} \rightarrow$ single

$N\sqrt{N} \rightarrow$ N inputs

Prime Factorisation for a Range

for any 'no'
Store the smallest p.f

N = 30

pfa

x	2	3	2	5	2	7	2	3	2	11	2	13	2	3	2	17	2	19	2	3	2	23	2	5	2	3	2	29	2
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

i^2

① Build a sieve storing the smallest pf at every i

② Prime factorisation

N = 24
↓ /2
12 /2
↓
6 /2 → 3 /3 → 1 stop

$2^3 \cdot 3 = 24$

21 → $\boxed{3}$

```
no = input()
while(no > 1){
    print ( pfa[no] ),
    no = no / pfa[no];
}
```

sieve $\rightarrow$ $N \log \log N$

$+ \underline{\underline{N}} \log_2 N$

$\uparrow$

for N Numbers

$\downarrow$

$7 \rightarrow 7$

$\downarrow$

$21 = \underline{3 \times 7}$

X

$X/6$

$X/7$

$X/8$

by same No ≥ 2

in

each step.

$100 \xrightarrow[100/7]{} 13 \rightarrow \cdots 1$

$100 \rightarrow 50 \rightarrow 25 \rightarrow 12 \rightarrow 6 \rightarrow 3 \rightarrow \underline{\underline{1}}$

Applications

=) Count the no divisors of 24

8 divisor

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

$\sqrt{N}$ Brute force

log N

$24 = 2^3 \cdot 3^1$

$\|(0,1,2,3)$

$(4).(2) = \boxed{8}$

$2 \times 3 = 6$

$2 \times 1 = 2$

$1 \times 3 = 3$

$8 \times 3 = 24$

$8 \times 1 = 8$

$\vdots$

$N = p_1^{\boxed{a_1}} \cdot p_2^{a_2} \cdots p_k^{a_k}$

$\text{Divisors} = (\underbrace{a_1+1}_{0 \text{ to } a_1})(\underbrace{a_2+1}_{0 \text{ to } a_2}) \cdots (a_k+1)$

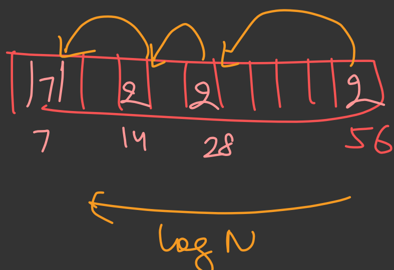
formula

$$(3+1)(1+1) = 4 \times 2 = 8 \text{ ways}$$

$$\underline{56} = \rightarrow \underline{2^{3} \cdot 7^{1}}$$ (Prime factorisation)

$$= \begin{array}{cc} \underline{2^0} & 7^0 \\ 2^1 & 7^1 \\ 2^2 & \\ 2^3 & \end{array}$$

= 1
= 7
= 2
= 14
= 4
= 28
= 8
= 56



5×3×2

$$30 = 2^{1} \cdot \underbrace{3^{1} \cdot 5^{1}}$$

$$= (1+1)(1+1)(1+1)$$

$$= 8$$

2^0 3^0 5^0

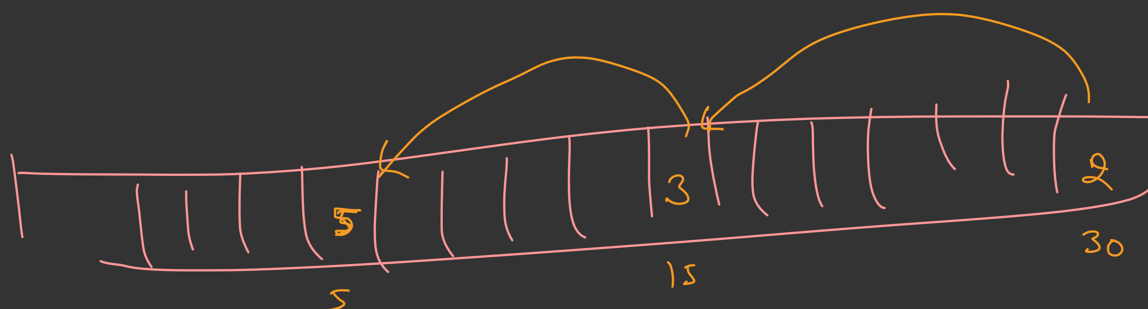
2^1 3^1 5^1

$2^0 = 1$ 1   1

$2^1 = 2$ 3   5

$= 1, 3, 15, 5$

$= 30, 10, 6, 2$



$30 = 2 \times 3 \times 5$