Day1

06 November 2022 18:25

1. Prime Number
2. LCM
3. GCD

* topics
  - Prime Number
    - Worst T.C → O(N)

Prime Number

n

↓

itself or 1 divisible

n = 2 → 1, 2

n = 3 → 1, 3

n = 4 → 1, 2, 4 ✗

n = 5 → 1, 5

Prime

n

1, n

2 to n-1

C++ code ⟶ Worst

n = 5 i = 2

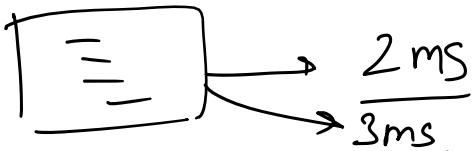
5 % 2 == 0 ✓

X

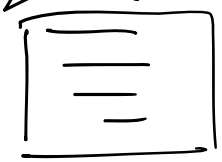
```
int n;
cin >> n;

for(int i=2; i<=n-1; i++){
    if(n % i == 0){
        print("False"); break;
    }
}
print("True");
```

T.C — O(N)

S1 (16GB, 256 SSD



S2 (16GB, 256 SSD)



3ms

Asymptotic Notations ⟶

Worst Case T.C

↓

Bigoh O(n)

T.C- O(1) O(S)

n = 23

1 × 23 = 23

Divisor of a Number always occur in pair.

n = 24

1 × 24 = 24

2 × 12 = 24

3 × 8 = 24

4 × 6 = 24

6 × 4 = 24

8 × 3 = 24

12 × 2 = 24

24 × 1 = 24

* Point to the value Repeat नहीं होना चाहिए

n = 24

$\sqrt{24} = \lfloor 4.6 \rfloor$

ceil(4.2) ⇒ 5

ceil(4.6) ⇒ 5

(-5)

i = 2 to $\sqrt{n}$

$i = 2$ to $\sqrt{n}$

$n \% i$

$24 \% 2 \rightarrow$

$n = 18$

$\sqrt{n} \Rightarrow 4.2$

(5)

2 to 5

False

$n = 23$

$\sqrt{23} = [4.5]$

(5)

2 to 5.

$23 \% 2$ ✗

$23 \% 3$ ✗

$23 \% 4$ ✗

$23 \% 5$ ✗

Prime

Algo

(1) ** Divisors always occur in pairs

(2) $i = 2$ to ceil(sqrt(n));

(3) if (n % i == 0) {

(3) if (n % i == 0) {
    print("Not a Prime"); break;
}

(4) At the end of the loop
~~print~~("Prime");

T.C - O(N)

n = 200000

(200000)

T.C - O(√n)

n = 200000

√n

450

Prime Number

2 or 3 ~~को~~ छोड़ कर

6n+1, OR 6n-1 n = 1 से (

2 ✗
3 ✗ 6n+1 , 6n-1
5 n=1 7 , 5
7 n=2 13 11

7 n=2 13 ✓ 11 ✓

11 n=3 19 ✓ 17 ✓

13

17 n=4 25 ✗ 23 ✓

Prime Number — T.C - $O(\sqrt{n})$

```
bool primeNumber(int n){
✓ if(n==2 || n==3) return true;
  if(n%2==0 || n%3==0) return false;
```

★ $(6n+1, 6n-1)$ $n=1 \Rightarrow 5$ $(i^2 \le (\sqrt{n})^2$ $i=2$ to sqrt(n)

★ Divisors always occur in pairs

```
for(int i=5; i*i<=n; i=i+6){
```

or $i \le sqrt(n)$ (5, 11)

```
  if(n%i==0 ✓
     ||
     ✓ n%(i+2)==0){
    return false;
  }
}
return true;
```

6

n = 67 n = 73

① $i^0 \times i^0 \leq n$

② $i = i + 6$

i = 5 67%5 ✗ 25 ≤ 67

i = 7 67%7 ✗ 49 ≤ 67

) $i^6 = i^0 + 6$ 121 ≤ 67

i = 11

n = 73

i = 5 73%5 ✓ 25 ≤ 73 ✓

73%7 ✗ 49 ≤ 73

i = 11 (121) < 73

return true;

① $i^0 \times i^0 \leq n$

② $i = i + 6$

③ n % i == 0

||

n % i+2 == 0

return false

i = 2 to 8

T.C - O(√n)

GCD

a = 4 b = 6

2 × 2
2 × 3 (2)

Euclid's Algorithm

4) 6 (

4) 4 (

4
X

Ans

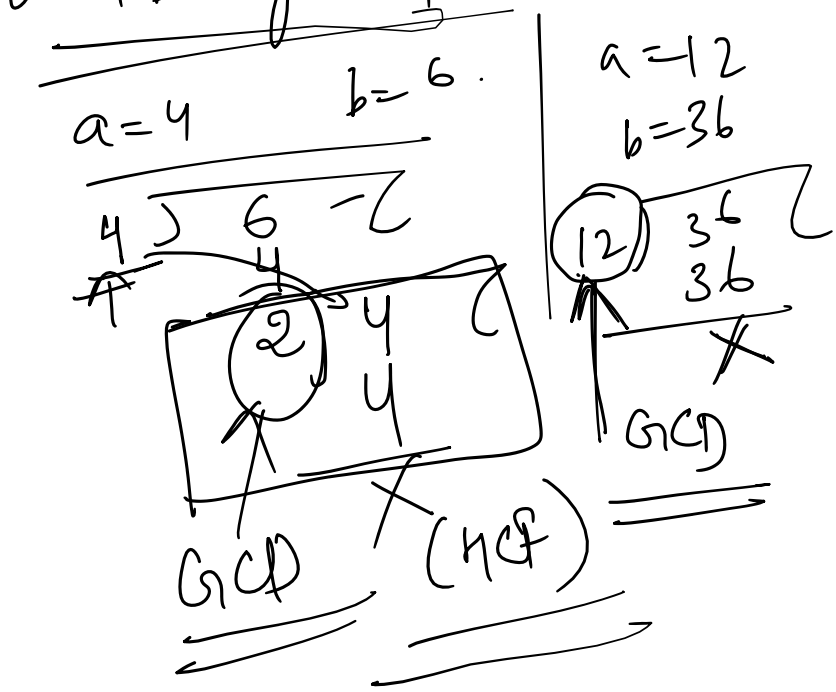
n = 12
m = 36

12 → {2 × 2 × 3}
36 → {2 × 2 × 3 × 3}

{2, 2, 3}

2+2+3 = 12

Euclid's Algorithm



Recursion

```
int GCD(int divisor, int dividend){
    if(dividend % divisor == 0)
        return divisor;
return GCD(dividend % divisor, divisor);
}
```

a=4 b=6

GCD(4,6) (Arr2)

6%4 = 2

4) 6 (
4
2) 4 (

6%4 = 2

GCD(2,4)

4%2 → 2

$O(\log(\min(a,b)))$

T.C —

n = 5 (1) m = 6 (1)

5) 6 (

5

1 | 5

5

X

GCD

m = 100

n = 3

3) 100 (

99

1 | 3

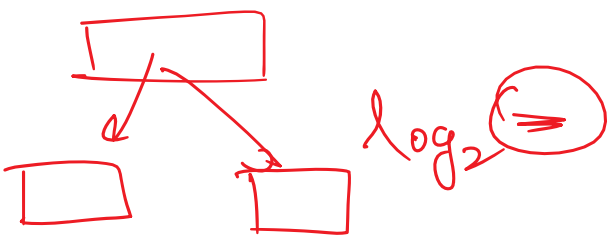
3

X

T.C - $O(\log(\min(a,b)))$

** जब भी चीज Divide. होती है

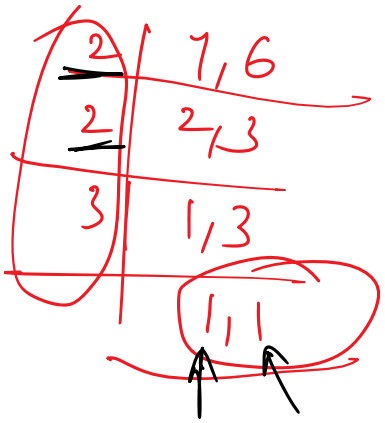
तो T.C $\log_2$



T.C - O(log(min(a,b)))

LCM

a = 4 b = 6



2 x 2 x 3

4 x 3 = 12

LCM

GCD

$a \times b = \gcd(a,b) \times lcm(a,b)$

$a=4 \quad b=6$

4 | 6 (
4
2 | 4 (
4
x

$2 \times 6 = 2 \times lcm(4,6)$

$12 = lcm(4,6)$

$T.C = O(\log c^{(\min(a,b))})$

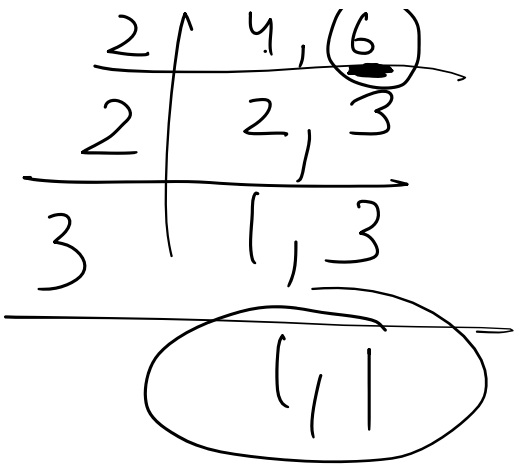
```
int LCM(int a, int b){
 int i=2; int ans=1;
 while(a!=1 && b!=1){
  if(a%i!=0 && b%i!=0)
     i++; continue;
  }
  if(a%i==0)
```

ans = 2 x 2 x 3 (12)

a = 4 b = 6

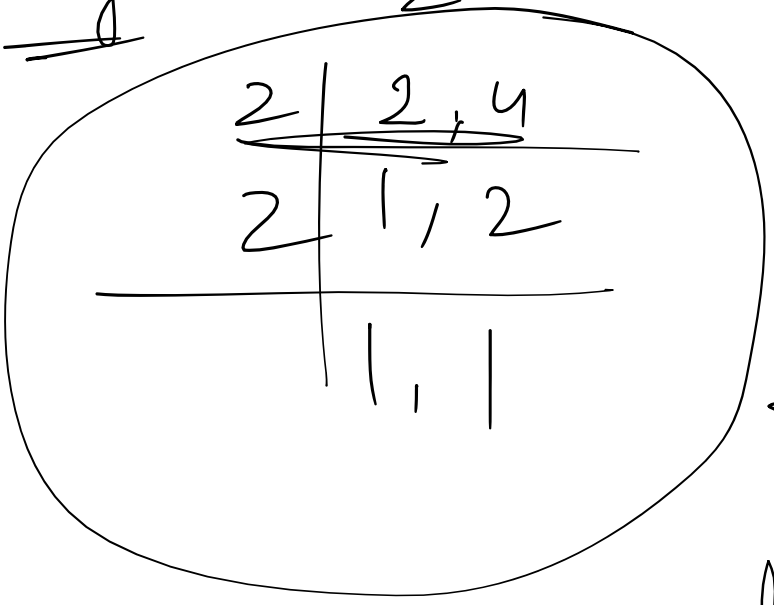
2 | 4, 6

```
if(a % i==0)
  { a=a/i;
  }
if( b % i==0)
  { b=b/i;
  }
ans *= i;
}
return ans;
}
```



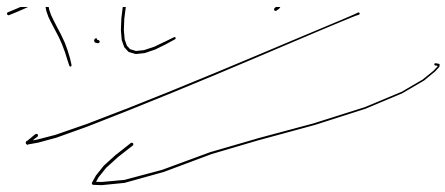
T.C $\max(a, b)$

eg $a = 2 \quad b = 4$



T.C $\max(a, b)$

T.C $\log c(\min(a, b))$



① GCD (2)