Today's Content

* Gcd
* Properties of GCD
* Gcd Optimization.
* Check if a given number is prime
* Find all primes in range [1 – N].

Gcd: Greatest Common Divisor

HCF: Highest Common Factor

$$gcd(a,b) = x \; \{ x \text{ is greatest no. s.t. } a \% x = 0 \;\&\&\; b \% x = 0 \}$$

$gcd(15,25) = 5$

15 = 1, 3, 5, 15

25 = 1, 5, 25

$gcd(12,30) = 6$

12 = 1, 2, 3, 4, 6, 12

30 = 1, 2, 3, 5, 6, 10, 15, 30

$gcd(10,-25) = 5$

10: 1, 2, 5, 10

-25: -25, -5, -1, 1, 5, 25

$gcd(0,8) = 8$

0: 1, 2, 3, 4 . 8 . ∞

8: 1, 2, 4, 8

$$\{ \boxed{gcd(0,x) = |x|}, \; x \neq 0 \}$$

$gcd(0,-10) = 10$

-10: -10, -5, -2, -1, 1, 2, 5, 10

$gcd(0,0)$: Undefined

1	1
2	2
3	3
4	4
⋮	⋮
∞	∞

Properties of Gcd

$gcd(a,b) = gcd(b,a)$ // Commutative

$$\left.\begin{aligned} gcd(a,b,c) = \; & gcd(gcd(a,b), c) \\ & gcd(gcd(a,c), b) \\ & gcd(gcd(b,c), a) \end{aligned}\right\}$$ // Associative property

Special Property

$\{ A, B > 0 \;\&\&\; A >= B \;\&\&\; gcd(A, B) = x \}$

$\hookrightarrow \boxed{gcd(A-B, B) = x}$

$\downarrow$

$A \% x = 0$

$B \% x = 0$

x is the highest factor.

$$gcd(\overset{A}{23}, \overset{B}{5}) \Rightarrow gcd(18,5) \Rightarrow gcd(13,5) = gcd(8,5) = gcd(3,5)$$

* We are doing repeated subtraction.

$$\therefore \; gcd(A, B) = gcd(A\% B, B)$$

$gcd(24, 16) \Rightarrow gcd(24\% 16, 16) \Rightarrow gcd(8, 16) \Rightarrow gcd(8, 16) \Rightarrow gcd(8, 16)$

(∞ - loop) ...

From properties: $\boxed{gcd(A,B)} = gcd(B, A) = gcd(A\% B, B)$

$\boxed{gcd(A,B) = gcd(B, A\% B)}$

(recursion)

$gcd(24,16) \Rightarrow gcd(16,8) \Rightarrow gcd(8,0) = \underline{\underline{8}}$

$gcd(14,24) \Rightarrow gcd(24,14) \Rightarrow gcd(14,10) = gcd(10,4) = gcd(4,2)$

$= gcd(2,0) = \underline{\underline{2}}.$

$gcd(0,10) \Rightarrow gcd(10,0) = \underline{\underline{10}}$

Pseudocode:

// Assumption: To calc & return gcd(a,b)

```
int gcd(int a, int b) {
    if (b==0) { return a }
    return gcd(b, a%b)
}

main() {
    gcd(|a|, |b|)
}
```

Code: ↓

Note: gcd(0,0) = 0

return according to Qn

TC: $O(\log_2(\min(a,b)))$

SC →

Proof of Gcd TC: $O(\log(\min(a,b)))$

First notice that for arbitrary integers x, y such that $x \geq y$, we get $x \% y \leq \frac{x}{2}$. This is due to:

- $x \% y < y$
- $x \% y \leq x - y$

During the process of the algorithm, suppose we have our current integers $a, b, a \geq b$. In the following 2 steps, both of them are being taken modulo a value not greater than them. Using the above observation, after these 2 steps we get 2 integers at most $\frac{a}{2}, \frac{b}{2}$ respectively. Once one of them becomes 0 the algorithm terminates, so the algorithm terminates in at most $2\log_2(\min(a,b))$, which implies $\mathcal{O}(\log(\min(a,b)))$.

In School:

gcd(14, 24)

```
14 ) 24 ( 1
     14
     ----
     10 ) 14 ( 1
          10
          ----
           4 ) 10 ( 2
                8
               ----
                2 ) 4 ( 2
                    4
                   ----
                    x
```

Prime Numbers : No. with only 2 factors (1 & itself)

↳ 2 different factors

5, 9, 11, 1

idea: Brute force

count factors → == 2: prime! / != 2: not prime!

```
bool isPrime(int n) {
    c = 0
    for (i=1; i<=n; i++) {
        if (n%i == 0) { c++ }
    }
    if (c==2) { return true }
    else { return false }
}
```

TC: $O(n)$

SC: $O(1)$

Optimize [Class 1 of intermediate]

N = 24

$i <= N/i$

$i * i <= N$

$i <= \sqrt{N}$

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

N = 25

i	<=	N/i	Factors
1		25	+2
5		5	+1

```
bool isPrime(int n) {
    c=0
    for(i=1; i*i<=n; i++){
        if(n%i == 0){
            c = c+2
            if (i == n/i){
                c--
            }
        }
    }
    if (c==2) { return true }
    else { return false }
}
```

TC: $O(\sqrt{N})$

SC: $O(1)$

Qn: Given N, find all prime no. from [1 - N].

N = 10 [1-10] : 2, 3, 5, 7

N = 20 [1-20] : 2, 3, 5, 7, 11, 13, 17, 19

Brute force idea: For all no. from 1-N, check if they are prime or not.

```
getAllPrimes (int n) {
  for (i = 1; i <= n; i++) {
      if ( isPrime (i)) {
          print (i)
      }
  }
}
```

TC: $O(N\sqrt{N})$

SC: $O(1)$

// Given N = 50, [1 - 50] get Primes.

1 F	2 T	3 T	4 F	5 T	6 F	7 T	8 F	9 F	10 F
11 T	12 F	13 T	14 F	15 F	16 F	17 T	18 F	19 T	20 F
21 F	22 F	23 T	24 F	25 F	26 F	27 F	28 F	29 T	30 F
31 T	32 F	33 F	34 F	35 F	36 F	37 T	38 F	39 F	40 F
41 T	42 F	43 T	44 F	45 F	46 F	47 T	48 F	49 F	50 F

whoever marked 4 as false, also marked its multiples as false.

Pseudocode:

```
getAllPrimes(int n){
  bool p[n+1] = {True}   // mark all no. as True.
  p[0] = p[1] = false
  for(i=2; i<=n; i++){
      if(p[i] == True){   // i is a prime no.
          for(j=2*i; j<=n; j=j+i){   // mark all
              p[j] = false            // multiples of i
          }                           // as false.
      }
  }
}
```

TC:

$i=2$	$i=3$	$i=4$	$i=5$	$i=7$ ...	$i=11$ ...	$i=p$
$\Rightarrow N/2$	$N/3$	x	$\frac{N}{5}$	$N/7$	$N/11$	N/p

p = largest prime $<= N$

$$\Rightarrow N\underbrace{\left(\frac{1}{2}+\frac{1}{3}+\frac{1}{5}+\frac{1}{7}\cdots+\frac{1}{p}\right)}_{\text{sum of reciprocals of primes}}$$

idea:

$$\left(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{N}\right)$$

$$\int_1^N \frac{1}{x} \Rightarrow \log N$$

TC $\Rightarrow O(N \log\log N)$, SC: $O(N)$

$N = 2^{64}$

$$\log(2^{64}) = 64$$

$$\hookrightarrow \log(64) \Rightarrow 6.$$

$\therefore \log(2^{64}) = 64$ / $\log(\log(2^{64})) = 6.$

$$\log(N) > \log(\log N)$$

2 → 4, 6, 8, 10 . . . .

3 → 6, 9, 12, 15 . . . .

5 → 10, 15, 20, 25 . . . .

7 → 7*2, 7*3, 7*4, 7*5, 7*6, 7*7

∴ directly start j from i*i

```
getAllPrimes(int n){
    bool p[n+1] = {True}   // mark all no. as True.
    p[0] = p[1] = false
    for(i=2; i<= √N ; i++) {
        if (p[i] == True) {   // i is a prime no.
            for(j = i*i ; j<=n; j=j+i) {   // mark all
                p[j] = false               // multiples of i
            }                              // as false.
        }
    }
}
```

When $i = 2$ — start: $j = 4$

$i = 3$ — $j = 9$

$i = 5$ — $j = 25$

. . . . .

$i = \sqrt{N}$ — $j = N$

$i = \sqrt{N}+1$? — $j = (\sqrt{N}+1)^2 > N$

$i = x$

$j = x * x = x^2$

$i = \sqrt{N}$

$j = N$

TC ⇒ $O(N \log \log N)$, SC: $O(N)$

Doubts: ?

Eg:- A: 1 2 3 4 5

B: 10

l	r	m		
1	5	3	→ ✗	go left.
1	2	1	✓	go right.
2	2	2	✓	go right
3	2	Stop! [l > r]		

A=5

2 : 2, 4, 6, 8, 10 . . . _

3 : 3, 6, 9, 12, 15 . . . .

2, 3, 4, 6, 8, 9, 10, 12, 14, 15 . . . .

1 – 10 : $10/2 = 5$ $-10/6 = 1$

$10/3 = 3$

$\underbrace{n/a + n/b}_{\text{Count of nos div by } [1-n]} - \dfrac{n}{lcm(a,b)}$

$5+3-1 = 7$.

$$\boxed{lcm(a,b) = \frac{a * b}{gcd(a,b)}}$$

$\dfrac{2 \times 3}{gcd(2,3) = 1} = 6$.

$a^m * a^n = a^{m+n}$

$N\sqrt{N}$

↳ $N^1 * N^{1/2}$

$N^{1+\frac{1}{2}} = N^{3/2}$