Math for DSA

Prime Numbers

1. A no. which is divisible by 1 & no. itself onl

⇒ Let's take an example of 36

1 x 36
2 x 18
3 x 12
4 x 9
6 x 6
9 x 4
12 x 3
18 x 2
36 x 1

⇒ Repeating

⇒ So, we don't need to check for values which are greater than sqrt(N).

⇒ Time Complexity ⇒ $O(\sqrt{N})$

⇒ only check for those no's who are less tho $\sqrt{n}$

1.0 Find all the Prime no.s Till 40

=> we can run a loop from 2 to 40 & check for each number.

=> Time Complexity:- $O(n\sqrt{n})$

=> Now, we know that 2 is prime, but multiples of 2 will not prime, similarly 3 is prime but multiples of 3 will not be prime.

=> now, if we eleminate multiples of prime no.s ~~of~~, we will get an optimised solution

Prime

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~~

31 ~~32~~ ~~33~~ ~~34~~ ~~35~~ ~~36~~ 37 ~~38~~ ~~39~~ ~~40~~

now If we repeat this process till sqrt(n) we will get all the prime no.s from 2 to 40

⇒ Time complexity

$$\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7} \cdots \frac{n}{p}$$

$$\Rightarrow n \left(\underbrace{\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} \cdots \frac{1}{p}} \right)$$

⇓

Harmonic Progression

(Tailor Series)

⇓

$\log(\log(N))$

Total time complexity ⇒ $O(N \times \log(\log(N)))$

2.8 Find Sqrt of a Number ?

⇒ To find sqrt what we can do is check from 1 to number, if $(\text{element})^2$ is == number or not

Time complexity ⇒ $O(N)$

⇒ Optimised approach

0 to 36

⇒ we have a range of numbers which is sorted, hence we can apply binary search here.

0 . . . 18 . . . 36

18*18 => 36

0 . . . 9 . . . 18

9×9 > 36

0 . . . 4 . . . 9

4×4 < 36

4 . . . 6 . . . 9

6×6 = 36

Answer = 36

Time complexity :- O(logn)

=> To get value in decimal upto p

=> Increament root with 0.1
check if root * root <= n

Repeat this process in forloop
~~upto~~ from 1 to p

Newton Raphson Method

$$\boxed{root = \frac{\left(x + \frac{N}{x}\right)}{2}}$$

where, x is assumed sqrt of N

⇒ why the formula works?

⇒ Let's say your assumed value for x is correct i.e $x = \sqrt{N}$

$$\sqrt{N} = \frac{\left(\sqrt{N} + \frac{N}{\sqrt{N}}\right)}{2}$$

$$\Rightarrow (\sqrt{N} + \sqrt{N})/2$$

$$\Rightarrow \frac{2\sqrt{N}}{2}$$

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

⇒ here, Error = $|root - x|$
Error should be minimum

⇒ Let's put, $x = n$

```
while True
    root = (x + n/x)/2
    if (|root - x|) < 1 )// error<1
        break;
    x = root
```

-1 Let's take an example of 40

$x = 40$

while True

root = (40 + 40/40)/2 = 20.5

Error = |root − x| = 40 − 20.5 = 19.5

(1) 19.5 > 1

x = root = 20.5

(2) root = (20.5 + 40/20.5)/2 =

root = (20.5 + 1.95)/2 = 11.225

(20.5 11.225) > 1

root = 11.225

(3) root = (11.225 + 40/11.225)/2 = 7.39

(11.225 7.39) > 1

root = 7.39

(4) root = (7.39 + 40/7.39)/2 = 6.40

7.39 − 6.40 = 0.99 < 1

Hence break

ans = 6.40

to get more precise value decrease the limit of error.

Factors of A Number

⇒ Factors of 20 ⇒ 1×20
2×10
4×5
5×4
10×2
20×1

⇒ Again Repeating

⇒ Now check roots only upto $\sqrt{n}$

Time complexity :- $O(\sqrt{n})$

```
if (i % n == 0)
    print (i + n/i);
```

Modulo Properties

i) $(a+b) \% m = ((a \% m) + (b \% m)) \% m$

ii) $(a-b) \% m = [(a \% m) - (b \% m)] \% m$

iii) $(a * b) \% m = [(a \% m) * (b \% m)] \% m$

iv) $\left(\frac{a}{b}\right) \% m = [(a \% m) * (b^{-1} \% m)] \% m$

$b^{-1} \% m$ ⇒ multiplicative modulo Inverse (mmi)

Ex 1.a $(6 * y) \% 7 = 1$

$y =$ mmi for 6 & $y = 6$

$(6 \times 6) \% 7 = 36 \% 7 = 1$

mmI $\Rightarrow$ means b & m are co-primes (only common factor = 1)

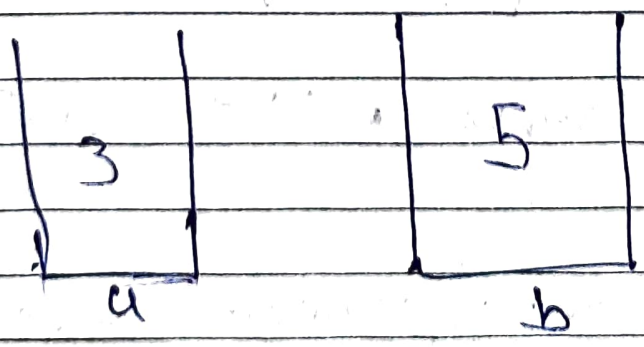
2. $(a \% m) \% m = a \% m$

vi) $m \% m = 0 \quad \forall x \in$ +ve Integers

NOM Extra: If p is prime no. which is not a divisor of b then $ab^{p-1} \% p = a \% p$ due to Fermat's Little Theorem.

Die - Hard Example

$\Rightarrow$ You are give to buckets which have capacity of 3 & 5 gallons respectively & you have get exactly 4 gallons of water.



$$\overset{a}{(0}, \overset{b}{6}) \to (3, 0) \to (0, 3) \to (3, 3)$$

$$(0, 4) \leftarrow (3, 1) \leftarrow (0, 1) \leftarrow (1, 6) \leftarrow (1, 5)$$

$(0, 4)$ ⇓ Ans

$(1, 5)$ ⇓ empty bucket

$$\text{Ans} = a s_1 - b s_2$$

where, s_1 = no. of times a is filled

&

s_2 = no. of times b is emptied

$$\boxed{3x + 5y = 4}$$

If $HCF(a, b) / L$ is an Integer it will work

Euclid's Algorithm

~~gcd (105~~

$\Rightarrow$ $\gcd(a, b) = \gcd(\text{rem}(a, b), a)$

$\gcd(105, 224) = \gcd(\text{rem}(224, 105), 105)$

$= \gcd(14, 105)$

$105x + 24y$

$\Downarrow$

$14x + 105y$

=> If (a == 0)
return b

gcd(b%a, a)

LCM (Least Common Factor)

=> lcm(a, b) => min no. divisible by both a & b

Note

Let a, b ~~to~~ two numbers

$d = gcd(a, b)$

$f = \frac{a}{d}$, $g = \frac{b}{d}$

=> $a = f \times d$ & $b = d \times g$

$lcm = l$

$lcm(a, b) = lcm(fd, gd)$

Now f & g have no common factor

$$lcm = f \times g \times d$$

$$a \times b = fd \times gd$$

$$= d\cancel{(f \times g)} \; d \times d\cancel{fg}$$

$$\boxed{a \times b = hcf \times lcm}$$

$$\boxed{lcm = \frac{a \times b}{HCF(a,b)}}$$