Todays Content:

→ isPrime

→ Prime Seive

→ Count factors

Oct 5th: Holi

Oct 24th: ___

Oct 26th: ___

IsPrime(): number with only 2 factrs

N=10 : Not prime

N= 7 : Yes

IsPrime(): Count of factrs of N = 2

↳ i=1...n & count factrs : TC: O(N)

↳ ≈→ i = 1... √n & count factrs : TC: O(√N)

i) Print all Primes from 1-N:

N=10 : output : 2 3 5 7

Idea1: iterate on all numbers from 2 ... N & check if number is prime & print

TC: $N * \{\underline{\sqrt{N}}\} \rightarrow O(N\sqrt{N})$

↳ TC for ease prime function

Idea2: Say we need all prime 1-50

	✓	✓	✗	✓	✗	✓	✗	✗	✗
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
✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
31	32	33	34	35	36	37	38	39	40
✓	✗	✓	✗	✗	✗	✓	✗	✗	✗
41	42	43	44	45	46	47	48	49	50

Pseudocode: Seive of Erthotheres / Prime Seive

```
void  allPrime(int n){
    bool p[N+1] = T → {Initialize all value, without iterations}
                       TODO
    p[0] = F   p[1] = F
    i = 2; i<= N; i++){
        // We iterate on multiple of i, if i is a prime
        if (p[i]==T){ // i is a prime
            // iterate on multiple of i till N ......
            j = 2*i; j<= N; j= j+i){
                // j is mul of i & it is not prime
                p[j] = F
            }
        }
    }
    i = 2; i<= N; i++){
        if(p[i]== True){ print(i)}
    }
}
```

i

→ 2*i } +i

3*i } +i

4*i } +i

5*i

⋮

TC: Given N:

i	j : mul of i till N	Itera
2	mul of 2 till N	N/2
3	mul of 3 till N	N/3
4	—	
5	mul of 5 till N	N/5
6	—	
7	mul of 7 till N	N/7
⋮		
N	mul of N till N	N/N

Total Iterations:

$$S = \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \cdots \frac{N}{N}$$

$$= N\left[\tfrac{1}{2} + \tfrac{1}{3} + \tfrac{1}{5} + \tfrac{1}{7} + \cdots \tfrac{1}{N}\right]$$

Sum of reciprocals of all prime till N $\approx \log_2(\log(N))$

TC: $N * [\log(\log N)]$ SC: $O(N)$

$N = 2^{32}$ $\log_2(N) = 32$

↳ $\log_2(\log N)) = \log_2(32) = 5$

Optimization to above idea :

10:40 → 10:50

N = 36

i	mut of i, if is prime					
2 →	2*2	2*3	2*4	2*5	2*6	2*7
3 →	3*2	3*3	3*4	3*5	3*6	3*7
4 →	*					
5 →	5*2 2	5*3 3	5*4 2	5*5	5*6	5*7
6 →	*					
7 →	7*2 2	7*3 3	7*4 2	7*5 5	7*6 2	7*7

```
void allPrime(int n){   T.C: O(N log(log N))   S.C: O(N)
    bool p[N+1] = T → { Initialize all value, without iterations }
                        TODO
    p[0] = F   p[1] = F
    i = 2; i*i <= N; i++){          → opt1
        // we iterate on multiple of i, if i is a prime
        if (p[i] == T){ // i is a prime
            j = i*i; j <= N; j = j+i){      opt2
                // j is mul of i & it is not prime
                p[j] = F
            }
        }
    }
    i = 2; i <= N; i++){
        if(p[i] == True){ print(i) }
    }
}
```

[Claim: last i value for which we will enter inner loop: $\sqrt{N}$]

$i : [2 \ N] \quad j = [i*i \to N]$

2

:

$\sqrt{N} \longrightarrow j = [N \to N]$ 1 iteration

$\sqrt{N}+1 \longrightarrow j = i*i > N$: 0 iterations

$j = [\sqrt{N}+1]^2 = N+1+2\sqrt{N} > N$

$\sqrt{N}+2 \to$ no iterations

Find no:of factrs fr all [1- N]

N=10 :	1	2	3	4	5	6	7	8	9	10
#facts :	1	2	2	3	2	4	2	4	3	4

Idea1: Fr all numbers from 1 - N iterate & calculate no:of factrs

O(N) O(√N)

TC: N√N

Idea2:

N=15.

✓	✓													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\boxed{1}$	$\boxed{2}$	$\boxed{2}$	$\boxed{3}$	$\boxed{2}$	$\boxed{4}$	$\boxed{2}$	$\boxed{4}$	$\boxed{3}$	$\boxed{4}$	$\boxed{2}$	$\boxed{6}$	$\boxed{2}$	$\boxed{4}$	$\boxed{4}$
1	1 2	1 3	1 2 4	1 5	1 2 3 6	1 7	1 2 4 8	1 3 9	1 2 5 10	1 11	1 2 3 4 6 12	1 13	1 2 7 14	1 3 5 15

```
int[]   All factrs (int n) {
    int fct[N+1] = {1}
    i=2 ; i<=N; i++) {
        //i is factr to all mul of i =  i  2i  3i  4i  5i....
        j=i ; j<=N; j=j+i) {
            //i is factr of j
            fct[j] = fct[j]+1        fct[j] = fct[j]+i
                                     sum of all factr of j
        }
    }
    return fct[]
}
```

Dryrun: N=8	1	2	3	4	5	6	7	8
fct[]	1	2	2	3	2	4	2	4

Table Given N

i	j : mul of i till N	iter
2	mul of 2 till N	N/2
3	mul of 3 till N	N/3
4	mul of 4 till N	N/4
5	mul of 5 till N	N/5
⋮		
N	mul of N till N	N/N

Total Iter =

$$S = \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \frac{N}{5} + \cdots \frac{N}{N}$$

$$\underline{N\left[\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots 1/N\right]}$$

sum of reciprocals of all natural numbers [2 N]

$\approx \log_2^N$

$$S = N * \log_2^N \Rightarrow \boxed{TC: O(N\log_2^N)}$$