Todays Content:

- isPrime
- Prime Seive
- Count factors
- spf[ ]
- Segmented Seive [Optimal Content]

IsPrime(): # factrs = 2

N = 10 : Not prime

N = 7 : prime

IsPrime(): Calculate no: of factrs

↳ TC1: O(N): 1.. N

↳ TC2: O(√N): 1.. √N

Print all Primes from 1- N:

N = 10: output: 2 3 5 7

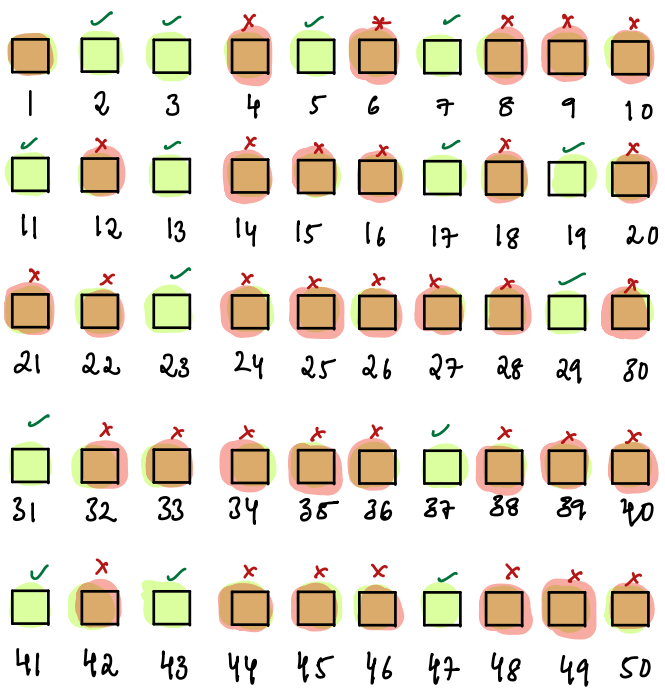
Idea1: Iterate from 1- N & check if every number prime n not

↳ O(√N)

TC: N * √N ≈ O(N√N)

↳ TC for isprime()

Idea2: Say we need all prime 1-50



Pseudo code: Seive of Eratosthenes a Prime Seive

```
void allPrime(int n){
    bool p[n+1] = {True} // 1-based indexing
    p[1] = false
    i = 2; i*i <= N; i++){
        //iterate m multiple of i, if i is a prime number
        if(p[i] == True){
            //multiple of i = 2i +i 3i +i 4i .....
            j = 2i; j*i<=n; j = j+i){
                // j is not prime
                p[j] = false
            }
        }
    }
    // iterate m p[] & get indices with True
}
```

TC: Given N:

i	j	Itera
2	mul of 2 till N	N/2
3	mul of 3 till N	N/3
4	x	
5	mul of 5 till N	N/5
6	x	
7	mul of 7 till N	N/7
8		
11	mul of 11 till N	N/11
:		
x		

Total Iterations:

: $\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \frac{N}{11} + \cdots$

$N\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \cdots\right]$

[// Sum of inverse of prime till N / log(log N)} maths: ref]

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

Optimization to above idea :

N = 25

1 mut of i

1 :

2 : 2*2 2*3 2*4 2*5 ....

3 : 3*2 3*3 3*4 3*5 ....

4 :

5 : 5*2 5*3 5*4 5*5 ....

6 :

7 : 7*2 7*3 7*4 7*5 7*6 7*7 ....

```
void allPrime(int n){   TC: Nlog(logN)   SC: O(N)
    bool p[n+1] = {True} //1-based indexing
    p[i] = false
    for(i=2; i*i<=N; i++){
        //iterate on multiple of i, if i is a prime number
        if(p[i] == True){
            //multiple of i = 2i, 3i, 4i .....
            for(j=i*i; j<=n; j=j+i){
                // j is not prime
                p[j] = false
            }
        }
    }
    // iterate on p[] & get indices with True
}
```

↳ TODO:
↳ Construct Table you will understand

Enter 2nd loop if

i =	: j = (i*i)
$\sqrt{N}$	N ✓
$\sqrt{N}+1$	$(\sqrt{N}+1)(\sqrt{N}+1)$ *
	$N+2\sqrt{N}+1 > N$
$\sqrt{N}+2$	*

Find no: of factors for all [1 - N]

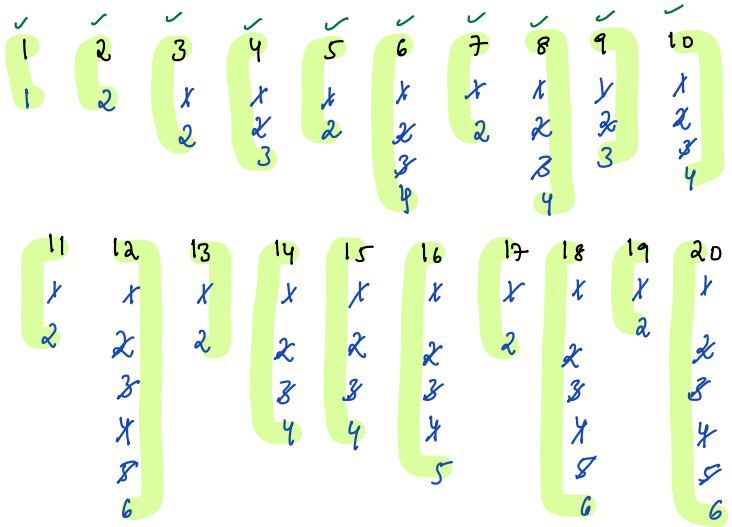
N = 10:	1	2	3	4	5	6	7	8	9	10
factors:	1	2	2	3	2	4	2	4	3	4

Idea1: for every number from [1 - N] find no: of factors

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

Idea2:

N = 20:



int[] All factors (int n) { Note: count of factors of N = f[N]

```
int f[n+1] = {1}; // Because for all numbers 1 is factor
f[0] = // no: of factors for 0, depend on question
i = 2; i <= n; i++) {          // i is a factor to itself
    // multiple of i = i, 2i, 3i, 4i ----
    j = i; j <= n; j = j + i) {
        // i is a factor of j, inc count by 1
        f[j] = f[j] + 1
    }
}
return f
}
```

Table Given N

i	j	iter
2	1st mul of 2	$N/2$
3	1st mul of 3	$N/3$
4	1st mul of 4	$N/4$
5	1st mul of 5	$N/5$
6	1st mul of 6	$N/6$
⋮		
N	1st mul of N	N/N

Total Iter =

$= N/2 + N/3 + N/4 + \cdots N/N$

$= N\left[1/2 + 1/3 + 1/4 + 1/5 + \cdots 1/N\right]$

= Sum of all reciprocals from [2 - N]

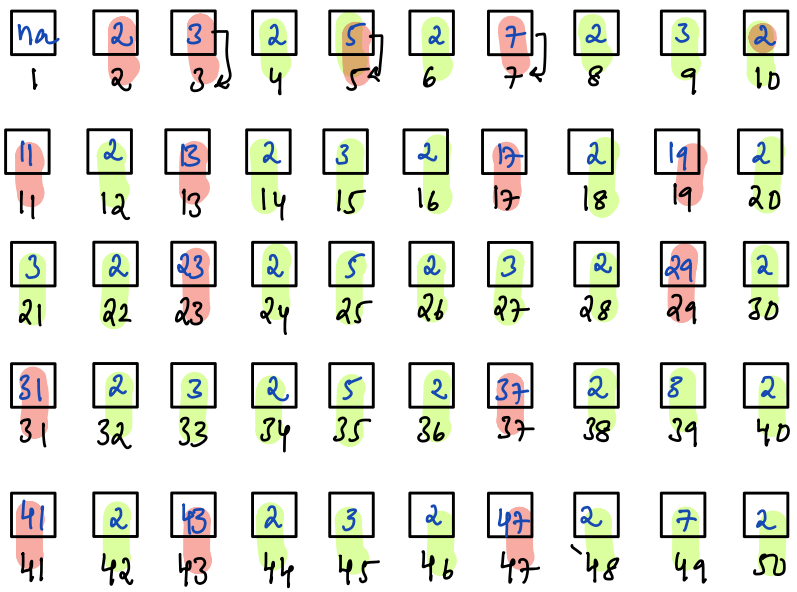
$= \log N$

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

Spf: Find Smallest prime factor for all numbers from 1- N

N=10 :	1	2	3	4	5	6	7	8	9	10
spf[]:	0	2	3	2	5	2	7	2	3	2

Idea: Say we need spf for all numbers from 1- 50



```
int[] AllSpf(int N){   TC: O(N + N log(log(N)))  SC: O(N)
    int spf[N+1] = {0}
    i=0; i<=N; i++){ → TC: O(N)
        // initialize spf[i] = i
    }
    i=2; i*i<=N; i++){ → TC: O(N log(log N))
        // i is a prime number? i can be spf to all mul of i
        if(spf[i]==i){ // goto mul of i
            j=i*i; j<N; j=j+i){
                spf[j] = min(i, spf[j])
            }
        }
    }
    return spf
}
```

i mul of i

2 : #spf=2 — 2*2 2*3 2*4 2*5 2*6 2*7 . . .

3 : 2 — 3*2 3*3 3*4 3*5 3*6 3*7 →

4 :

5 : 2 — 5*2 3 — 5*3 2 — 5*4 5*5 5*6 5*7 . . . →

6 :

7 : 2 — 7*2 3 — 7*3 2 — 7*4 5 — 7*5 2 — 7*6 7*7 - - - →

Properties:

If N is a prime it will have no factrs [2 $\sqrt{N}$]

for N: To get no: of factrs

```
for(i=1; i*i <= N; i++){
    if(N%i==0){
        if(i == N/i){
            c = c+1
        }
        else{ c = c+2}
    }
}
```

N = 7

i	i*i <= 7	i	N/i	c = 0
1	✓	1	7	c = c+2
2	✓	2		
3	*	return 2		

N = 23

i	i*i <= 23	i	N/i	c = 0
1	✓	1	23	c = c+2
2	✓	not factr		
3	✓	not factr		
4	✓	not factr		
5	*	return 2		

Claims:

If N is a prime it will have no factrs [2 $\sqrt{N}$]

If N is not prime it will have atleast 1 factr [2 $\sqrt{N}$]

↓

1 prime

Segmented Sieve:

Given [a, b] find no: of primes in range [a, b]

Constraints:

$1 <= a <= b <= 10^9$

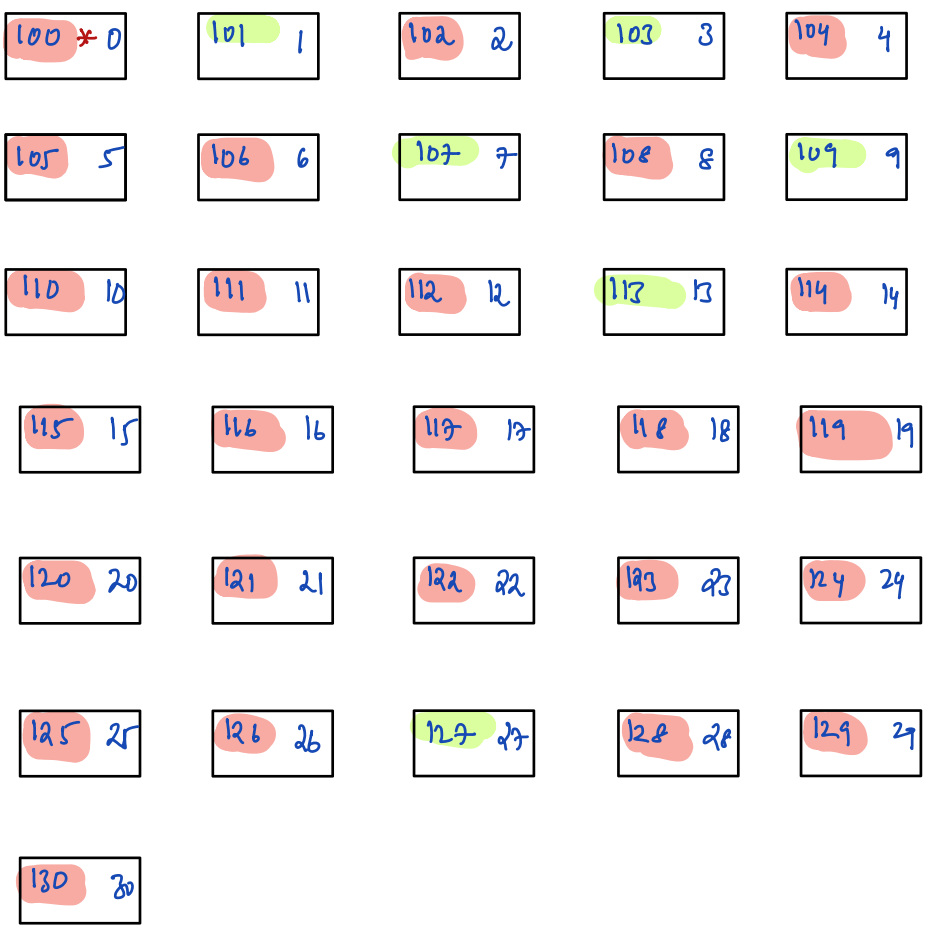
$1 <= b - a <= 10^5$

Note: In general max array size we can declare is $10^5 - 10^6$

Idea: Get all primes from [1 - b] using sieve

TC: $b \log(\log(b))$ SC: $O(b)$ → $b = 10^9 \rightarrow p[10^9]$ ✗

Ex: a = 100, b = 130, sqrt(130) = 11 = {2 3 5 7 11} bool p[31]



1st mul of i : [a = 100, b = 120]

1st mul of 2 : 100

1st mul of 3 : $\left[\frac{100}{3}\right] * 3 + 3$ = 33 * 3 = 99 + 3 = 102

1st mul of 5 : 100

1st mul of 7 : $\left[\frac{100}{7}\right] * 7 + 7$ = 98 + 7 = 105

1st mul of 11 : $\left[\frac{100}{11}\right] * 11 + 11$ = 99 + 11 = 110

1st mul of i : if (a % i == 0) { 1st mul = a }

else { 1st mul = $\left[\frac{a}{i}\right] * i + i$ }

```
int SegPrime(int a, int b){

    //Step:1 → Get all primes from [1, √b] & store in list
        ↳ Solve this using sieve
        ↳ //Say we are store only prime in list prime
        list<int> primes // All prime all [1, √b]

    //Step:2 → ch[b-a+1] = True

    i = 0; i< primes.size(); i++){

        p = prime[i]
        //Get 1st mul of p in [a b]
        if(a%p==0){ mul = a}
        else{ mul = (a/p)*p + p}
        if(mul == p){ mul = mul + p}
        j = mul; j<=b; j=j+p){
            ch[j-a] = false
        }
    }
}
```

Edge Case: no: of primes [a 1, b 50] : {2, 3, 5, 7}

All primes [1 √50] = 1 7

[a 1, b 50]

Actual 1st

1st mul of 2 : $[a/2]*2+2 = [1/2]*2+2 = 0+2 = 2 \rightarrow 4$ (0)

1st mul of 3 : $[a/3]*3+3 = [1/3]*3+3 = 0+3 = 3 \rightarrow 6$

1st mul of 5 : $[a/5]*5+5 = [1/5]*5+5 = 0+5 = 5 \rightarrow 10$

1st mul of 7 : $[a/7]*7+7 = [1/7]*7+7 = 0+7 = 7 \rightarrow 14$

$1 \le a \le b \le 10^9$

$1 \le b - a \le 10^5$

Step 1: List of all primes $[1 - \sqrt{10^9}]$ = ✓

Step 2: $ch[b - a + 1] = ch[10^5]$ = ✓