- Introduction to Prime Numbers
- Get all primes 1 to n
- Smallest Prime Factor for 2 to n
- Prime Factorization
- Get the no. of factors / divisors

Prime Numbers

Numbers that have 2 factors → 1 and itself.

Eg:- 2, 3, 5, 7, 11...

Q1) Given a number n, check if it is prime or not.

$n = 3 \longrightarrow$ True

$n = 6 \longrightarrow$ False

```
boolean checkPrime(n){
    count = 0
    for(i=1; i*i<=n; i++){
        if(n%i == 0){
            count++
            if(i != n/i){
                count++
            }
        }
    }
    return (count == 2)
}
```

Alt. approach

```
if(n<2)
    return false

for(i=2; i*i<=n; i++){
    if(n%i == 0)
        return false
}
return true
```

$O(\sqrt{n})$ T.C

$O(1)$ S.C.

Q2) Given n, print all the prime nos. from 1 to n.

$n = 10 \rightarrow 2, 3, 5, 7$

$n = 20 \rightarrow 2, 3, 5, 7, 11, 13, 17, 19.$

```
void printAllPrime(n){
    for(i-> 2 to n){
        boolean isPrime = true
        for(j=2; j*j<=i; j++){
            if(i%j==0){
                isPrime = false
                break
            }
        }
        if(isPrime)
            print(i)
    }
}
```

$\sqrt{2} + \sqrt{3} + \sqrt{4} + \dots + \sqrt{n}$

$= O(n\sqrt{n})$ T.C.

$O(1)$ S.C.

Sieve of Eratosthenes

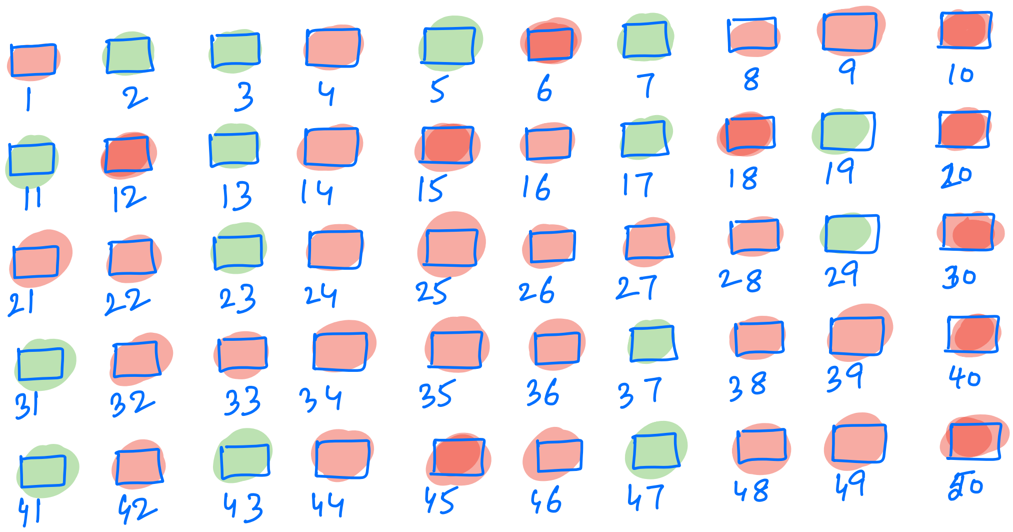
→ Assume that all nos, 2 to n are prime

→ Start with the first prime number and mark all its multiples as non-prime.

→ Move to the next unmarked nos, (not marked in red), this is prime, mark all its bigger multiples as non-prime.

→ Repeat the process till the end of array

→ All unmarked nos. are prime.



i	# iterations
2	$n/2$
3	$n/3$
4	$\cancel{\ }$
5	$n/5$
6	$\cancel{\ }$
7	$n/7$
8	$\cancel{\ }$

$$\frac{n}{2}+\frac{n}{3}+\frac{n}{5}+\frac{n}{7}+\frac{n}{11}+\frac{n}{13}+\dots$$

$$= n\underbrace{\left(\frac{1}{2}+\frac{1}{3}+\frac{1}{5}+\frac{1}{7}+\frac{1}{11}+\frac{1}{13}+\dots\right)}_{\text{Sum of reciprocals of prime nos. upto } n.}$$

$$= O(n\,(\boxed{\log(\log n)}))\ \text{T.C.}$$

```
printAllPrimes(n){
    boolean isPrime[n+1] = {true}
    isPrime[0/1] → false
    for(i=2; i*i<=n; i++){
        if(isPrime[i]){
            for(j= i*i; j<=n; j+=i){
                isPrime[j] = false
            }
        }
    }
    for(i=2; i<=n; i++){
        if(isPrime[i]){
            print(i)
        }
    }
}
```

$O(n\log(\log n))$ T.C

$O(n)$ S.C

7

$7*2 \rightarrow 2$

$7*3 \rightarrow 3$

$7*4 \rightarrow 2$

$7*5 \rightarrow 5$

$7*6 \rightarrow 2,3$

$7*7 \rightarrow$

$7*8$

$7*9$

```
for(i=2; i<=a; i++){
    if(isPrime[i]){
        for(j=i*2; j<=b; j++){
            ...
        }
    }
}
```

$i=2 \Rightarrow j \rightarrow b/2$

$i=3 \Rightarrow j \rightarrow b/3$

$i=5 \Rightarrow j \rightarrow b/5$

$$\frac{b}{2}+\frac{b}{3}+\frac{b}{5}+\cdots+\frac{b}{\sim a}$$

$$= b\left(\frac{1}{2}+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{a}\right)$$

$$= b\log(\log a)$$

$a=n \quad b=n$

$n\log(\log n)$

$a=\sqrt{n}, b=n$

$n\log(\log\sqrt{n}) = O(n\log(\log n))$

[Break till 10:30 PM]

Q3) Given n, return the smallest prime factor for each nr. from 2 to n.

n = 10	2	3	4	5	6	7	8	9	10
Ans →	2	3	2	5	2	7	2	3	2

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

$\left.\begin{array}{c}\boxed{i}\\ i\end{array}\right\}$ Prime Nr.

```
fn smallestPrimeFactor(n){
    int spf[n+1]
    for(i -> 2 to n)
        spf[i] = i
    for(i = 2; i*i <= n; i++){
        if(spf[i] == i){
            for(j = i*i; j <= n; j += i){
                if(spf[j] == j)
                    spf[j] = i
            }
        }
    }
    return spf
}
```

$O(n\log(\log n))$ T.C

$O(n)$ S.C

Prime Factorization

$n = 48$

$$\begin{array}{r|l} 2 & 48 \\ 2 & 24 \\ 2 & 12 \\ 2 & 6 \\ 3 & 3 \\ & 1 \end{array}$$

$2^4 * 3^1$

$2^0, 2^1, 2^2, 2^3, 2^4$ $\quad$ $3^0, 3^1$

$5 * 2 = 10.$

$48 = 2 * 2 * 2 * 2 * 3$

$= 2^4 * 3^1$

Nr. of divisors $= (4+1) * (1+1)$

$= 5 * 2 = 10.$

1, 2, 3, 4, 6, 8, 12, 16, 24, 48.

10 divisors.

$2^{0-4} * 3^{0-1}$

$2^0 * 3^0 = 1$

$2^0 * 3^1 = 3$

$2^1 * 3^0 = 2$

$2^1 * 3^1 = 6$

$2^2 * 3^0 = 4$

⋮

$n = 300$

$$\begin{array}{r|l} 2 & 300 \\ 2 & 150 \\ 3 & 75 \\ 5 & 25 \\ 5 & 5 \\ & 1 \end{array}$$

$300 = 2^2 * 3^1 * 5^2$

$2^{0-2} * 3^{0-1} * 5^{0-2}$

divisors $= (2+1) * (1+1) * (2+1)$

$= 18.$

$n = 560$

$$\begin{array}{r|l} 2 & 560 \\ 2 & 280 \\ 2 & 140 \\ 2 & 70 \\ 5 & 35 \\ 7 & 7 \\ & 1 \end{array}$$

$560 = 2^4 * 5^1 * 7^1$

$d \longrightarrow 2^{\leq 4} * 5^{\leq 1} * 7^{\leq 1}$

$(4+1) * (1+1) * (1+1) = 5 * 2 * 2 = 20.$

$$n = p_1^{a_1} * p_2^{a_2} * p_3^{a_3} * \dots p_k^{a_k}$$

Primes

$$\Rightarrow \boxed{\#\text{ factors of } n = (a_1+1)*(a_2+1)*(a_3+1)*\dots*(a_k+1).}$$

$$20 = 2^2 * 5^1$$

$$(2+1)*(1+1) = 6 \text{ factors.}$$

$n = 360$

$$\boxed{2^{\cancel{2}\,3} \;*\; 3^{\cancel{1}\,2} \;*\; 5^1}$$

$\text{spf}[360] = 2$

$\downarrow /2$

$\text{spf}[180] = 2$

$\downarrow /2$

$\text{spf}[90] = 2$

$\downarrow /2$

$\text{spf}[45] = 3$

$\downarrow /3$

$\text{spf}[15] = 3$

$\downarrow /3$

$\text{spf}[5] = 5$

$\downarrow /5$

1

$$(3+1)*(2+1)*(1+1)$$

$$= 4*3*2 = 24.$$

```
fn factors(n) {
    ans = 1
    s = spf[n]
    while(n>1){
        cnt = 0
        while(n%s==0){
            n = n/s
            cnt += 1
        }
        ans = ans * (cnt+1)
        s = spf[n]
    }
    return ans
}
```

$O(\log n)$ T.C. after computing spf[].

$O(n * \log(\log n))$ for calculating spf $+$ $O(\log n)$ for finding # factors of n.

Find count of factors for m diff. values, each upto 10^6, $m \rightarrow 10^5$.

$[O(max * \log(\log max)) + m * O(\log max)]$

for calculating spf

for finding # factors of n.

$= O(max \cdot \log(\log(max)) + m \cdot \log(max))$ T.C

Using sqrt approach $\downarrow$ $O(m * \sqrt{max})$ T.C