Today's Agenda

- Introduction to Prime Numbers
- Get all primes from 1 to N
- Print smallest prime factor for 2 to N
- Prime Factorization
- Get the number of factors/divisors

what are prime numbers?

↓ numbers having only 2 factors $(N>0)$

1 → ✗

2 → 1, 2 ✓

5 → 1, 5 ✓

7 → 1, 7 ✓

11 → 1, 11 ✓

Ques. Check prime?

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

T.C → $O(\sqrt{N})$

S.C → $O(1)$

Ques 2) Given a number N, print all prime numbers from 1 to N.

N = 10, => 2, 3, 5, 7.

N = 20, => 2, 3, 5, 7, 11, 13, 17, 19.

Brute force :-

T.C → $N\sqrt{N}$

S.C → O(1).

```
for (i=2; i<=n; i++) {
    if (checkPrime(i)) {
        print(i);
    }
}
```

Optimized Approach :-

N = 50, [1 50]

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

```
getallPrimes (N) {
    bool P[N+1] = {T};
    P[0] = P[1] = f;
    for ( i = 2; i <= n; i++) {
        if (P[i] == True) {
            for (j = 2i ; j <= n; j = j+i) {
                P[j] = false;
            }
        }
    }
}
```

2 → 4, 6, 8, 10 . . . .

3 → 6, 9, 12, 15, 18 . . .

5 → 10, 15, 20, 25 . . .

7 → 7×2, 7×3, 7×4, 7×5, 7×6, 7×7 : . .

11 → 11×2, 11×3, 11×4, 11×5, - . . . 11×11

↪ Sieve of Eratosthenes.

getallPrimes (N) { ← Correct Code.

```
getallPrimes (N) {
    bool P[N+1] = {T};
    P[0] = P[1] = f;
    for ( i = 2; i <= √n; i++) {
        if (P[i] == True) {
            for (J = i*i ; J <= n; J = J+i) {
                P[J] = false;
            }
        }
    }
}
```

outer loop   inner loop

2 → $\simeq$ $N/2$

3 → $\simeq$ $N/3$

5 → $\simeq$ $N/5$

T.C → $\underbrace{\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7}} + \dots$

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

Sum of all reciprocals of prime numbers.

T.C $\rightarrow$ $O(\underline{N\log(\log N)})$ $\rightarrow$ $N = 2^{64} \simeq 10^{18}$

S.C $\rightarrow$ $O(N)$

$N\log\log 2^{64}$

$N\log 64$

$N \times 6$

$\Rightarrow 10^{18} \times \underline{6}$

Previous idea

$N\sqrt{N}$

$10^{18} \times \sqrt{10^{18}}$

$\Rightarrow \underline{10^{27}}$.

$2^{10} \simeq 1000$

$2^{60} \simeq (1000)^6$

$\underline{2^{60} \simeq 10^{18}}$.

Break 7:56 Am - 8:06 Am

Ques   Smallest Prime factor :-

Given N, return the smallest prime factors for all numbers from 2 to N

N = 10,

	2	3	4	5	6	7	8	9	10
spf →	2	3	2	5	2	7	2	3	2

for Prime no. → spf is the same number

~~1 / 1~~	2 / 2	3 / 3	4 / ~~4~~ 2	5 / 5	6 / ~~6~~ 2	7 / 7	8 / ~~8~~ 2	9 / ~~9~~ 3	10 / ~~10~~ 2
11 / 11	12 / ~~12~~ 2	13 / 13	14 / ~~14~~ 2	15 / ~~15~~ 3	16 / ~~16~~ 2	17 / 17	18 / ~~18~~ 2	19 / 19	20 / ~~20~~ 2
21 / ~~21~~ 3	22 / ~~22~~ 2	23 / 23	24 / ~~24~~ 2	25 / ~~25~~ 5	26 / ~~26~~ 2	27 / ~~27~~ 3	28 / ~~28~~ 2	29 / 29	30 / ~~30~~ 2

Edge :- if something is already m.p, don't mark it again.

```
— spf creation (N) {
      spf[N+1]  // initialize spf[i] = i .
      for (i=2; i<=√N; i++) {
          if (spf[i] == i) {    // prime check
              for (j=i*i; j<=n; j+=i) {
                  if (spf[j] == j) {  // unmarked
                      spf[j] = i
                  }
              }
          }
      }
}
```

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

$S.C \to O(N)$

Prime factorization :-

↳ Representing a no', multiples of powers of unique prime no's.

no. of factors :-

2	48
2	24
2	12
2	6
3	3
	1

$n=48 \Rightarrow 2^4 \times 3^1 \Rightarrow (4+1)(1+1) \Rightarrow \underline{10}$.

$n=45 \Rightarrow 3^2 * 5^1 \Rightarrow (2+1)*(1+1) \Rightarrow \underline{6}$.

$n=300 \Rightarrow 2^2 * 3^1 * 5^2 \Rightarrow (2+1)(1+1)(2+1)$
$\Rightarrow \underline{18}$.

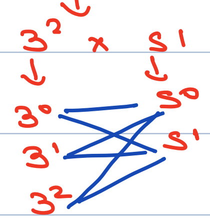
if you have a number N, whose prime factorization is,

$$p_1^{a_1} * p_2^{a_2} * p_3^{a_3} * \ldots p_y^{a_y}$$

no. of factors $\Rightarrow (a_1+1)*(a_2+1)\cdots(a_y+1)$

$n=45 \Rightarrow \underline{3^2 * 5^1} \Rightarrow (2+1)*(1+1) \Rightarrow \underline{6}$.

$\underline{1}, \underline{3}, \underline{5}, \underline{9}, \underline{15}, \underline{45}$



Ques

Given a number N. For all the numbers from 1 to N, get the number of factors/divisors

$N=10$. →

1	2	3	4	5	6	7	8	9	10
1	2	2	3	2	4	2	4	3	4

Brute force :- for all numbers from 1 to n, find count of factors by $\sqrt{n}$ method.

$O(N\sqrt{N})$.

Optimized.

$N=49$, $\frac{\cancel{49}^{7}}{\cancel{7}} \Rightarrow \frac{\cancel{7}^{1}}{\cancel{7}} \Rightarrow 1 \longrightarrow 7^2$

count of factors ⇒ 3.

$N=48$, $\frac{\cancel{48}^{24}}{\cancel{2}} \Rightarrow \frac{\cancel{24}^{12}}{\cancel{2}} \Rightarrow \frac{\cancel{12}^{6}}{\cancel{2}} \Rightarrow \frac{\cancel{6}^{3}}{\cancel{2}} \Rightarrow \frac{\cancel{3}^{1}}{\cancel{3}} \Rightarrow 1$

$2^4 \times 3^1 \Rightarrow \underline{10}$

count of factors

```
// create spf array first; -> nlog log n
for (i = 2; i <= n; i++) {  -> n log n
                                                      log n
        hm<int, int> ;
        x = i
        while (x > 1) {
            if (spf[x] is in hm) {
                hm[spf[x]] += 1
            }
            else {
                hm[spf[x]] = 1
            }
            x = n/spf[x] ;
        }

        // with the hm, you can calculate
           count of factors.
}
```

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

S.C → $O(n) + O(\log n)$

$O(n)$ ↓ spf array

$O(\log n)$ ↓ map.