

# Maths: Prime Numbers

## Agenda

- Intro to Prime no.
- Ect all primes from 1 to N
- No. of factors
- Question

Prime Number : Numbers that have exactly 2 factors

ex: 2, 3, 5, 7, ...

Check if a given integer is prime?

$n = 11$                       true

$n = 21$                       false

Solution : count # factors

code

```
bool checkPrime (N) {
```

```
    count = 0
```

```
    for (i = 1; i * i <= N; ++i) {
```

```
        if (N % i == 0) {
```

```
            if (i == N/i)    count++
```

```
            else            count += 2
```

```
        }
```

```
    }
```

```
    if (count == 2)
```

```
        return true
```

```
    return false
```

```
}
```

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

$SL = O(1)$

## Scenario

Seurk Prime wants to use random prime numbers from 1 to N.

$N = 10 \Rightarrow 2, 3, 5, 7$

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

## Brute force

```
for (i = 2 to N) {  
    if (checkPrime(i)) {  
        print(i)  
    }  
}
```

TC =  $O(N^2)$

SC =  $O(1)$

## Sieve of Eratosthenes

class teacher  $\rightarrow$  Souil  $\rightarrow$  distribute chocolates to  
prime roll no.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	✓	✓		✓		✓				✓		✓		
	T	T	<del>T</del>	T	<del>T</del>	T	<del>T</del>	<del>T</del>	<del>T</del>	T	<del>T</del>	T	<del>T</del>	<del>T</del>

$5 \times 2 = 10$

$5 \times 3 = 15$

$5 \times 4 = 20$

$5 \times 5 = 25$

...

code

```
bool isPrime [N+1]
```

```
for (i = 2; i <= N; i++) isPrime[i] = true
```

```
isPrime[1] = false
```

```
for (i = 2; i <= N; i++) {
```

```
    if (isPrime[i]) {
```

```
        for (j = i*i; j <= N; j += i) {
```

```
            isPrime[j] = false
```

```
        }
```

```
    }
```

```
}
```

$SL = O(N)$

```
for (i = 2 to N) {
```

```
    if (isPrime[i]) {
```

```
        print(i)
```

```
    }
```

Time complexity

i	j	iterations
2	4, 6, 8, 10, ...	$\sim N/2$
3	9, 12, 15, ...	$\sim N/3$
4	—	0
5	25, 30, ...	$\sim N/5$

...

...

$$\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \dots + \frac{N}{\sqrt{N}}$$

$$< \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \frac{N}{5} + \dots$$

$$= \sum_{i=2}^{\sqrt{N}} \frac{N}{i} = N \sum_{i=2}^{\sqrt{N}} \frac{1}{i}$$

$$\int \frac{1}{x} dx = \log x$$

$$TC < O(N \log N)$$

$$\text{exact} \rightarrow O(N \log(\log N))$$

$$\begin{array}{lll} N=10^6 & \log N = 20 & \log \log N = 4 \\ N=10^{12} & \log N = 40 & \log \log N = 5 \end{array}$$

## Question

Given a prime integer  $N$ , count # factors for all numbers from 1 to  $N$ .

	1	2	3	4	5	6
$N=6$	1	2	2	3	2	4
#factors						

Bruteforce : count factors for each no. from 1 to N

TC to calculate factors =  $O(\sqrt{N})$

total TC =  $O(N\sqrt{N})$

### Observation

factors of a number K is count of integers from 1 to K which divides K.

	✓ 1	✓ 2	✓ 3	✓ 4	✓ 5	✓ 6	✓ 7	✓ 8	✓ 9	✓ 10
N = 10										
# factors	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1	<del>0</del> 1
		2	2	2	2	2	2	2	2	2
				3		3		3	3	3
						4		4		4
	1	2	2	3	2	4	2	4	3	4

we can also skip 1 & start factor count = 1

Code

```
int factors[N+1]
```

```
for(i, factors[i] = 1
```

```
for(i=2; i<=N; ++i) {
```

```
    for(j=i; j<=N; j+=i) {
```

```
        factors[j]++ // i is a factor of j
```

```
    }
```

```
}
```

$$\text{iterations} = \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \dots$$

$$TC = O(N \log N)$$

$$SC = O(N) \text{ or } O(1) \text{ (if factors is output)}$$

# Sorted Permutation Rank

What is the rank of a given string w.r.t sorted order of its permutations. ↗ distinct chars.

eg "acb"

a b c	1
a c b	2
b a c	3
b c a	4
c a b	5
c b a	6

ans = 2

"bac" < "cab"

total permutations  $\rightarrow N!$

"date"

a \_ \_ \_  $\Rightarrow 3! = 6$

d a e \_  $\Rightarrow 1! = 1$

d a t e

$\Rightarrow 6 + 1 + 1 = 8$



$N=5$

0 1 2 3 4  
t r u n k  
✓ ✓ ✓

k \_ \_ \_ \_  $\Rightarrow 4! = 24$   
n \_ \_ \_ \_  $\Rightarrow 4! = 24$   
r \_ \_ \_ \_  $\Rightarrow 4! = 24$

#chars smaller than 't'  
 $3 \times 4!$

t k \_ \_ \_  $\Rightarrow 3! = 6$   
t n \_ \_ \_  $\Rightarrow 3! = 6$

$2 \times 3!$

t r k \_ \_  $\Rightarrow 2! = 2$   
t r n \_ \_  $\Rightarrow 2! = 2$

$2 \times 2!$

t r u k \_  $\Rightarrow 1! = 1$

⋮  
⋮  
⋮

t r u n k

Code

int fact[N]

fact[0] = 1

for (i = 1 to N-1) {

fact[i] = (fact[i-1] \* i) % M

}

int ans = 1

for (i = 0 to N-1) {

preprocess  
factorials

cnt = 0  $\rightarrow$  # chars smaller than s[i]

for (j = i+1 to N-1) {

if (s[j] < s[i])

cnt++

}

ans = (ans + (cnt \* fact[N-i-1]) % M) % M

}

return ans

TC =  $O(N^2)$

SC =  $O(N)$

factorial array