

Prime Numbers

Content

- Introduction to prime numbers
 - Get all primes from 1 to N
 - Print smallest prime factor from 2 to N
 - Prime factorization
 - Get the number of factors/divisors.
-

Criteria for personal referral \longrightarrow PSp $> 95\%$
Attendance $> 95\%$.

last time
 $\sim 67\%$

current PSp
65%

\longrightarrow 70%

What are prime numbers ?

positive numbers with exactly 2 factors

Q> Given a no. check whether its prime or not ?

Count the no. of factors

If 2 \rightarrow the no. is prime

else no. is not prime.

boolean checkPrime (int n) {

if (n < 2) {
 return false
}

count = 0

for (i = 1 ; i * i <= n ; i++) {

 if (n % i == 0) {
 if (i == n / i) {
 count += 1

 }
 else {
 count += 2

 }
}
return (count == 2)

TC: $O(\sqrt{n})$

SC: $O(1)$

Given a number N , we need to print all prime no. from 1 to N .

$N = 10 \rightarrow 2 \quad 3 \quad 5 \quad 7$

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

Brute force $\rightarrow \forall i_2^N \text{ checkPrime}(i) \xrightarrow{\text{true}} \text{print.}$

Tc: $O(N\sqrt{N})$

Sc: $O(1)$

Sieve of Eratosthenes

Class Prof \rightarrow Gowtham.

Abhishek \rightarrow class monitor

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

T ~~T~~ T ~~T~~ ~~T~~ ~~T~~ T ~~T~~ ~~T~~ ~~T~~ ~~T~~

~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~ ~~F~~

$$2 * 2 = 4 \quad F$$

$$2 * 3 = 6 \quad F$$

⋮

$$5 * 2 = 10$$

$$5 * 3 = 15$$

$$5 * 4 = 20$$

$$5 * 5 = 25$$

$$3 * 2 = 6 \quad F$$

$$3 * 3 = 9 \quad F \quad \checkmark \text{ start here}$$

$$7 * 7 = 49 \quad F$$

current prime = i

iterate till $i * i \leq N$

Pseudocode

SC: $O(N)$

boolean[] sieve = new boolean[N+1]

for (i → 0 to N) sieve[i] = true

sieve[0] = false

sieve[1] = false

for (i = 2 ; $i * i \leq N$; i++) {

 if (sieve[i]) {

 for (j = $i * i$; $j \leq N$; $j += i$) {

 sieve[j] = false

 }

 }

}

Time Complexity

i	j	# iterations.
2	2*2, 2*3, 2*4...	$\sim \frac{N}{2}$
3	3*3, 3*4.....	$\sim \frac{N}{3}$
4	_____	0
5	5*5, 5*6	$\sim \frac{N}{5}$

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

$$N \left\{ \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \dots \right\}$$

sum of reciprocals of prime no.
 $\log(\log N)$

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

Break 22:17

Given N , return the smallest prime factor for all numbers 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

do not update 12 \because 2 is spf.

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

Pseudocode

SC: $O(N)$

`int[] spf = new int[N+1]`

`for (i → 0 to N) spf[i] = i`

`spf[0] = -1`

`spf[1] = -1`

`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`

}

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

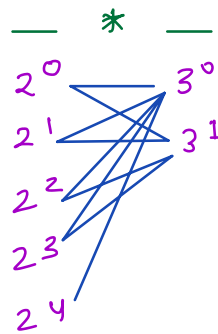
NOTE : How to solve for dynamic values ?

A = 13 45 68 56 19 42

→ create `spf[69]`

Basic Prime Factorization

$$\begin{array}{ll}
 2 & 48 \\
 2 & 24 \\
 2 & 12 \\
 2 & 6 \\
 3 & 3 \\
 & 1
 \end{array}
 \longrightarrow
 \begin{array}{l}
 2 * 2 * 2 * 2 * 3 \\
 = 2^4 * 3^1
 \end{array}$$

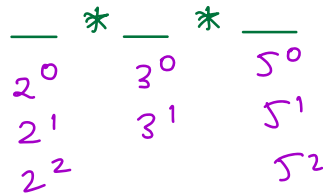


10 factors

1
3
2
6
4
12
8
24
16
48

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

$$2^2 * 3^1 * 5^2$$



overall
18 factors.

$$N = p_1^{a_1} * p_2^{a_2} * p_3^{a_3}$$

$$\# \text{ of factors} = (a_1 + 1) * (a_2 + 1) * (a_3 + 1)$$

Given N , For all the no. from 1 to N .
print # of factors or divisor

$N = 10$

	1	2	3	4	5	6	7	8	9	10
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Ans	1	2	2	3	2	4	2	4	3	4
	{1, 3}	{1, 2, 3}	{1, 3}	$\left\{\frac{1}{2}, \frac{1}{4}\right\}$	$\left\{\frac{1}{5}\right\}$	$\left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right\}$	$\left\{\frac{1}{7}\right\}$	$\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}\right\}$	$\left\{\frac{1}{3}, \frac{1}{9}\right\}$	$\left\{\frac{1}{2}, \frac{1}{5}, \frac{1}{10}\right\}$

Bruteforce

$ans[N+1] = [0] \dots$

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

for $i \rightarrow 1$ to N {
 $ans[i] = \text{countFactor}(i)$ } TC: $O(\sqrt{N})$
 }

We spf to optimize

1	2	3	4	5	6	7	8	9	10
1	2	3	2	5	2	7	2	3	2
11	12	13	14	15	16	17	18	19	20
11	2	13	2	3	2	17	2	19	2
21	22	23	24	25	26	27	28	29	30
3	2	23	2	5	2	3	2	29	2
31	32	33	34	35	36	37	38	39	40
31	2	3	2	5	2	37	2	3	2

u1	u2	u3	u4	u5	u6	u7	u8	u9	50
u1	2	u3	2	3	2	u7	2	7	2

$$A[i] = 48$$

2	48
2	24
2	12
2	6
3	3
	1

$$2^4$$

$$am = 1$$

$$*(4+1)$$

$$3^1$$

$$*(1+1)$$

$$= \underline{\underline{10}}$$

3	9 9 9
3	3 3 3
3	1 1 1
3 7	3 7
	1

$$3^3$$

$$am = 1$$

$$*(3+1)$$

$$3 7^1$$

$$*(1+1)$$

$$= \underline{\underline{8}}$$

$$am = 1$$

2	2 10
3	1 0 5
5	3 5
7	7
	1

$$2^1$$

$$*(1+1)$$

$$3^1$$

$$*(1+1)$$

$$5^1$$

$$*(1+1)$$

$$7^1$$

$$*(1+1)$$

$$\underline{\underline{16}}$$

Pseudocode

// Create the spf array } $O(N \log \log N)$
spf[N+1]

TC: $O(N \log N)$ {
 any[N+1]
 for (i \rightarrow 1 to N) {
 any[i] = cntFacBySPF(i, spf)
 }
 print(any)

int cntFacBySPF (int n, int[] spf) {

 any = 1

 while (n > 1) {

 pow = 0

 sp = spf[n]

 while (n % sp == 0) {

 pow += 1

 n = n/sp

 }

 any *= (pow + 1)

 }

 return any

}

$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \dots 1$

TC: $O(\log N)$

SC: $O(1)$

Overall TC : $O(N \log N)$ sc : $O(N)$

Follow up \longrightarrow Given dynamic array

A = 51 76 94 210 999

\longrightarrow Create `spf[1000]`
follow exactly the above code.