

Теория чисел

Теория чисел

12 марта

- Проверка на простоту
- Факторизация
- Решето Эратосфена
- Решето Эратосфена с факторизацией

19 марта

Применение алгоритмов

26 марта

Командная тренировка

Теория чисел



Теория чисел изучает **целые числа и их свойства**.

- **Факторизация**
- **Решето Эратосфена**
- НОД и НОК
- Числа Фибоначчи
- Операции по модулю
- Бинарное возведение в степень
- ...

Проверка на простоту

Простое число — это натуральное число, имеющее ровно два различных натуральных делителя: 1 и само себя.

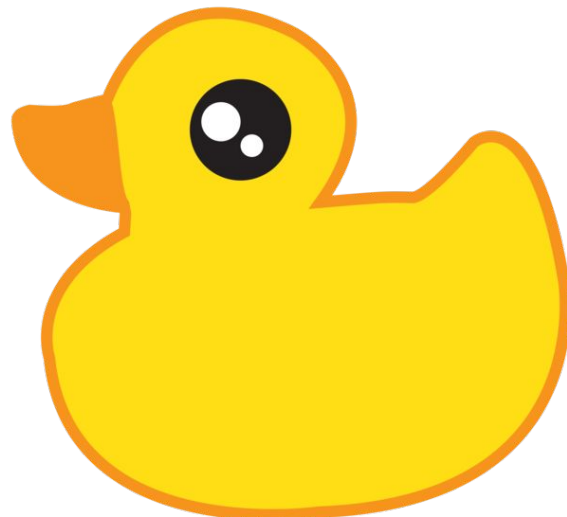
Составное число — это натуральное число, имеющее более двух различных натуральных делителей?

Пример:

5 – простое число?

12 – простое число?

1 - простое число?



Проверка на простоту



1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, ...

Проверка на простоту



1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, ...

Проверка на простоту

A decorative graphic consisting of three horizontal bars of different colors (teal, blue, yellow, and red) arranged in a staggered, overlapping fashion, creating a modern, abstract look.

Формулировка задачи:

Задано натуральное число X . Необходимо проверить, является ли оно простым.

Проверка на простоту



Формулировка задачи:

Задано натуральное число X . Необходимо проверить, является ли оно простым.

Можем проверить все натуральные числа от 2 до $X-1$ являются ли они делителем числа X . Если X делится на любое число из этого диапазона, то X — простое.

Проверка на простоту

```
for(int i = 2; i < n; ++i){  
    if(n % i == 0){  
        cout << "NOT PRIME";  
    }  
}  
cout << "PRIME";
```



Проверка на простоту

```
for(int i = 2; i < n; ++i){  
    if(n % i == 0){  
        cout << "NOT PRIME";  
    }  
}  
cout << "PRIME";
```



Асимптотика: $O(n)$
Что будет при $n = 10^9$?

Проверка на простоту

```
long long sqrt_n = round(sqrt((double)n));  
cout << sqrt_n << endl;  
for(long long i = 2; i <= sqrt_n; ++i){  
    if(n % i == 0){  
        cout << "NOT PRIME";  
        return 0;  
    }  
}  
cout << "PRIME";
```



Проверка на простоту

```
long long sqrt_n = round(sqrt((double)n));  
cout << sqrt_n << endl;  
for(long long i = 2; i <= sqrt_n; ++i){  
    if(n % i == 0){  
        cout << "NOT PRIME";  
        return 0;  
    }  
}  
cout << "PRIME";
```



Асимптотика: $O(\sqrt{n})$
Что будет при $n = 10^9$?

Проверка на простоту

```
for(long long i = 2; i*i <= n; ++i){  
    if(n % i == 0){  
        cout << "NOT PRIME";  
        return 0;  
    }  
}  
cout << "PRIME";
```



Асимптотика: $O(\sqrt{n})$

Факторизация



Факторизацией натурального числа называется его разложение на произведение простых множителей.

Пример: $40 = 2 * 2 * 2 * 5$

Задача:

Вам задано целое положительное число n . Разложите данное число на простые множители.

Факторизация

```
for(long long i = 2; i*i <= n; ++i){  
    while(n % i == 0){  
        cout << i << " ";  
        n = n / i;  
    }  
}  
if(n != 1)  
    cout << n << endl;  
return 0;
```



Асимптотика: $O(\sqrt{n})$

Решето Эратосфена



Решето Эратосфена – это алгоритм, который позволяет быстро найти все простые числа на отрезке $[1; n]$.



Решето Эратосфена



Решето Эратосфена – это алгоритм, который позволяет быстро найти все простые числа на отрезке $[1; n]$.

Запишем в ряд все числа от 2 до n .

Вычеркнем все числа, делящиеся на 2, кроме самого числа 2

Вычеркнем все числа, делящиеся на 3, кроме самого числа 3

Число 4 уже вычеркнуто

Вычеркнем все числа, делящиеся на 5, кроме самого числа 5

...

Решето Эратосфена

1,	2,	3,	4,	5,	6,	7,	8,	9,	10,
11,	12,	13,	14,	15,	16,	17,	18,	19,	20,
21,	22,	23,	24,	25,	26,	27,	28,	29,	30,
31,	32,	33,	34,	35,	36,	37,	38,	39,	40,
41,	42,	43,	44,	45,	46,	47,	48,	49,	50,
51,	52,	53,	54,	55,	56,	57,	58,	59,	60,
61,	62,	63,	64,	65,	66,	67,	68,	69,	...

Решето Эратосфена

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, ...

Решето Эратосфена

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, ...

Проверка на простоту

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, ...

Решето Эратосфена



```
int n;  
vector<char> prime (n+1, true);  
prime[0] = prime[1] = false;  
for (int i=2; i<=n; ++i)  
    if (prime[i])  
        if (i * 1ll * i <= n)  
            for (int j=i*i; j<=n; j+=i)  
                prime[j] = false;
```

Асимптотика: $O(n \log \log n)$

Решето Эратосфена с факторизацией



Идея:

Давайте вместо информации о простоте числа будем хранить его минимальный простой делитель.

Если число p – простое, $\text{sieve}[p] = p$.



Решето Эратосфена с факторизацией

```
const int N = 1000000;  
vector<int> sieve(n: N + 1);  
  
for (int i = 2; i <= N; ++i) {  
    if (sieve[i] == 0) {  
        sieve[i] = i;  
        for (int j = i * 2; j <= N; j += i) {  
            if (sieve[j] == 0) {  
                sieve[j] = i;  
            }  
        }  
    }  
}
```



Асимптотика: $O(n \log \log n)$

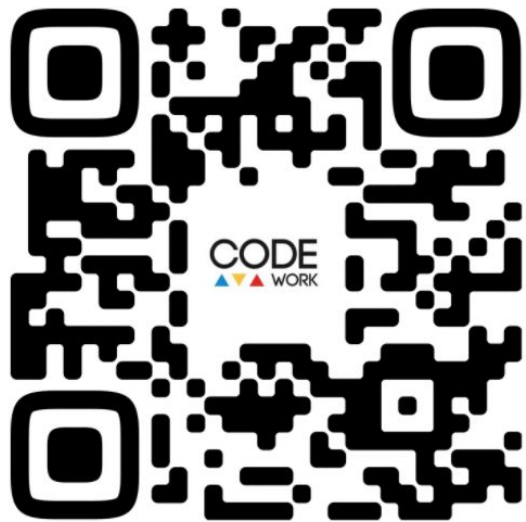
Решето Эратосфена с факторизацией

```
const int N = 1000000;  
vector<int> sieve(n: N + 1);  
  
for (int i = 2; i <= N; ++i) {  
    if (sieve[i] == 0) {  
        for (int j = i; j <= N; j += i) {  
            if (sieve[j] == 0) {  
                sieve[j] = i;  
            }  
        }  
    }  
}
```



Асимптотика: $O(n \log \log n)$

Заключение



Группа в VK



Div. A



Div. B