

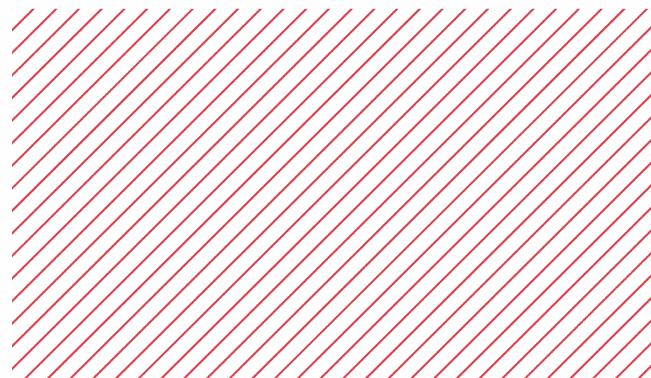
академия  
больших  
данных



# Базовая теория чисел

Артем Васильев

Алгоритмы и структуры данных Advanced





# Операция взятия остатка по модулю

---

- Представим число  $a = b * q + r$ , где  $b > 0$ ,  $0 \leq r < b$
- Такое представление единственно
- $q$  - “частное”
- $r$  - “остаток”
- Отношение эквивалентности на множестве целых чисел
- $a = b \pmod{m} \leftrightarrow (a - b)$  делится на  $m$



# Операция взятия остатка по модулю

---

- Выражается с помощью операции “%” в большинстве ЯП
- Осторожно с отрицательными числами!
- Java, C++:  $-5 / 2 = -1$
- Python:  $-5 \% 2 = 1$

# Факторизация числа

---

- Задача: найти все делители числа
- Наивное решение:  $O(n)$
- Решение за  $O(\sqrt{n})$ :

```
for (int i = 2; i * i <= n; i++) {  
    if (n % i == 0) {  
        // i and n/i are divisors  
    }  
}
```



# Подсчет количества делителей

---

- Если известна факторизация  $n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$
- Ответ это  $(k_1 + 1)(k_2 + 1) \dots (k_m + 1)$
- Можно быстрее!
- Поделим на все делители  $n$ , которые не больше  $n^{1/3}$
- Осталось три варианта:
  - a.  $n$  - простое
  - b.  $n = p^2$  для простого  $p$
  - c.  $n = pq$ , где  $p$  и  $q$  - различные простые



# Функция Эйлера

---

- Функция Эйлера  $\phi(n)$  (totient function) = количество остатков, взаимно простых с  $n$
- $\phi(n)$  - мультипликативная функция,  $(\phi(nm) = \phi(n) \phi(m)$  для любых взаимнопростых  $n$  и  $m$ )
- $\phi(n) = (p_1^{k_1} - p_1^{k_1-1}) (p_2^{k_2} - p_2^{k_2-1}) \dots$
- Считается, что подсчет не сложнее факторизации (RSA)



# Решето Эратосфена

---

- Как найти все простые числа от 1 до M?
- Поддерживаем массив `isPrime[i]` - известно ли, что число i простое
- Идем по i от 2 до M
- Если встретили простое, то пометим все кратные ему, как составные
- Время работы?



# Решето Эратосфена

---

```
isPrime[0] = isPrime[1] = false;
for (int i = 2; i < M; i++) {
    if (isPrime[i]) {
        for (int j = i * 2; j < M; j += i) {
            isPrime[j] =false;
        }
    }
}
```



# Решето Эратосфена

---

- Не хуже, чем  $n/2 + n/3 + n/4 + \dots + n/n = O(n \log n)$
- $k$ -е простое порядка  $k \ln k$
- Сумма будет порядка  $O(n \log \log n)$



# Линейное решето

---

- Для каждого числа  $i$  будем поддерживать  $lp[i]$  - минимальный простой делитель  $i$
- Также поддерживаем список всех найденных простых чисел  $P$
- Основная идея в том, чтобы выставить значение  $lp[i]$  ровно однажды для каждого числа
- Перебираем  $i$  по возрастанию, если  $lp[i] = 0$ , то число простое, добавляем его в список
- Перебираем все простые  $p$ , что  $p \leq lp[i]$ , ставим  $lp[p * i] = p$



# Линейное решето

---

```
for (int i = 2; i < M; i++) {
    if (lp[i] == 0) {
        lp[i] = i;
        primes.add(i);
    }
    for (int p : primes) {
        if (p > lp[i] || i * p >= M) {
            break;
        }
        lp[i * p] = p;
    }
}
```



# Алгоритм Евклида

---

- Находит наибольший общий делитель (GCD) двух чисел
- X делит A и B тогда и только тогда, когда X делит B - A и A
- Аналогично,  $B \% A$  и A
- $\text{gcd}(0, B) = B$
- $\text{gcd}(A, B) = \text{gcd}(B \% A, A)$



# Алгоритм Евклида

---

- $B \% A < B / 2$
- Поэтому, произведение чисел  $A$  и  $B$  уменьшается как минимум в 2 раза за один шаг
- Время работы  $O(\log A + \log B)$
- Худший случай  $A = F_n$ ,  $B = F_{n+1}$  (соседние числа Фибоначчи)



# Расширенный алгоритм Евклида

---

- Любая линейная комбинация  $X * A + Y * B$  делится на  $\gcd(A, B)$
- Существуют такие целые  $X$  и  $Y$ , что  $X * A + Y * B = \gcd(A, B)$
- Возвращаем тройку чисел  $(\gcd, X, Y)$  вместо только  $\gcd$
- Пусть мы знаем, что  $X * (B \% A) + Y * A = \gcd$
- Как получить такое же выражение для  $A$  и  $B$ ?



# Расширенный алгоритм Евклида

---

- $B \% A = B - (B / A) * A$
- $X * (B - (B / A) * A) + Y * A = \text{gcd}$
- $(Y - (B / A) * X) * A + X * B = \text{gcd}$
- Возвращаем тройку  $(\text{gcd}, Y - (B / A) * X, X)$
- Время работы все также  $O(\log A + \log B)$



# Линейные диофантовы уравнения

---

- Если  $A$  и  $B$  взаимно просты, то существует решение  $X * A + Y * B = 1$
- Значит, для любого  $C$  существует решение *линейного диофантового уравнения*  $X * A + Y * B = C$
- В общем случае,  $C$  должно делиться на  $\gcd(A, B)$



# Поиск обратного по модулю

---

- Складывать, вычитать, умножать по модулю - просто
- Как делить?
- Поделить = умножить на обратный
- *Обратным* к  $A$  элементом по модулю  $M$  называется такое число  $A^{-1}$ , что  $A * A^{-1} = 1 \pmod{M}$
- Существует тогда и только тогда, когда  $A$  и  $M$  взаимно просты



# Теорема Эйлера

---

- Используя функцию Эйлера, можно иначе найти обратный по модулю
- $a^{\phi(n)} = 1 \pmod{n}$ , если  $a$  и  $n$  взаимно просты
- Малая теорема Ферма - частный случай при простом  $n$
- Это значит, что  $a^{-1} = a^{\phi(n) - 1} \pmod{n}$
- Можно использовать двоичное возведение в степень
- Также  $O(\log n)$ , если знать  $\phi(n)$



# Теорема Эйлера (доказательство)

---

- Рассмотрим все остатки по модулю  $n$ , которые взаимно просты с  $n$
- Перемножим их все, обозначим произведение  $S$
- Домножим каждый остаток на  $a$ , получилась перестановка
- Опять возьмем произведение, получим  $a^{\phi(n)} S$
- Произведения равны, сократим на  $S$

# Решение линейных уравнений mod M

---

- Задача: найти такое  $x$ , что  $A * x = B \pmod{M}$
- В терминах линейного диофантового уравнения  $A * x = B + M * k$ ,  
или же  $A * x - M * k = B$ .
- Известны  $A, B, M$ , найти  $x, k$ .
- Можно, если  $B$  делится на  $\gcd(A, M)$
- Аналогично, можно решить  $(A / g) * x = B / g \pmod{M / g}$  путем  
нахождения обратного



# Китайская теорема об остатках

---

- Задача: даны два сравнения по модулю, найти  $x$ 
  - $x = a_1 \pmod{m_1}$
  - $x = a_2 \pmod{m_2}$
- Простой случай:  $m_1$  и  $m_2$  взаимно просты
- Существует единственное решение по модулю  $m_1 m_2$

# Китайская теорема об остатках

---

- Решим упрощенную задачу:
  - $x \equiv 1 \pmod{m_1}$
  - $x \equiv 0 \pmod{m_2}$
- Это значит, что  $x = m_1 p + 1$  и  $x = m_2 q$
- $m_2 q - m_1 p = 1$ , снова линейное диофантово уравнение!
- Или же  $m_2 q \equiv 1 \pmod{m_1}$ , и  $q \equiv m_2^{-1} \pmod{m_1}$
- Назовем решение  $x_1$



# Китайская теорема об остатках

---

- Аналогично решим
  - $x = 0 \pmod{m_1}$
  - $x = 1 \pmod{m_2}$
- Назовем решение  $x_2$
- Тогда нужный  $x = a_1 x_1 + a_2 x_2 \pmod{m_1 m_2}$
- Можно обобщить на  $n$  уравнений, объединяя два в одно



# КТО (не взаимно простые модули)

---

- Что произойдет, если  $m_1$  и  $m_2$  не взаимно просты?
- Иногда не существует решения:
  - $x \equiv 9 \pmod{15}$
  - $x \equiv 2 \pmod{24}$
- Если решение существует, то оно единственno по модулю  
 $\text{НОК}(m_1, m_2)$
- Сведите к линейному диофантовому уравнению