Числовые алгоритмы

Сложение

N - число, $n = \lceil \log N \rceil$ - длина

$O(n)$

Умножение

$O(n^2)$ $O(n^{\log_2 3})$ Карацуба $O(n \log n)$ БПФ

```
Mult (a, b):
    if b == 0:
        return 0
    res = Mult (a, ⌊b/2⌋) · 2
    if b mod 2 == 1:
        res = res + a
    return res
```

Деление

```
Div ( a, b)
    if a < b:
        return (0, a)
    (q, r) = Div (⌊a/2⌋, b)
    q = q · 2
    r = r · 2
    if a mod 2 == 1:
        r = r + 1
    if r ≥ b:
        r = r - b
        q = q + 1
```

$O(n^2)$

return (q, r)

Возведение в степень

```
pow(a, k):
    if k == 0:
        return 1
    res = pow(a, ⌊k/2⌋)
    res = res · res
    if k mod 2 == 1:
        res = res · a
    return res
```

~~$O(n^3)$~~

$$a = 2^n \qquad k = 2^n$$

$$a^k = (2^n)^{2^n} = 2^{n \cdot 2^n}$$

$$|a^k| = n \cdot 2^n$$

$$O(n^2 \cdot 2^{2n})$$

Модульная арифметика

≡ вычисления в полях mod M

$$|M| = n$$

Сложение

$$c = a + b$$

$$O(n)$$

```
if c ≥ M:
    c = c - M
```

Умножение

$O(n^2)$

$$Mult_M(a, b) = Div(Mult(a, b), M).r$$

Деление $O(n^3)$

$$Div_M(a,b) = Mult_M(a, b^{-1})$$

Возведение в степень

$$a^k = pow_M(a,k) \qquad O(n^3)$$

$Mult_M$ вместо $Mult$

Расширенный алгоритм Евклида

$$НОД(a,b) = НОД(b, a \% b)$$

$$\underset{?}{x} \cdot a + \underset{?}{y} \cdot b = НОД(a,b)$$

$a \% b$

$$d = x \cdot b + y \cdot (a - \lfloor a/b \rfloor \cdot b)$$
$$= \underbrace{y}_{x'} \cdot a + \underbrace{(x - \lfloor a/b \rfloor)}_{y'} \cdot b$$

```
EGCD(a, b):
    if b == 0:
        return (a, 1, 0)        # x, y
    (d, x, y) = EGCD(b, a % b)
    return (d, y, x - ⌊a/b⌋)
```

$O(n^3)$

Вычисление обратного

$EGCD(M, b) \rightarrow$

$x \cdot M + y \cdot b = d \mod M$ (x → 0)

Если b обратимо $\Rightarrow d = 1$

y: $y \cdot b = 1 \Rightarrow b^{-1} = y$

Числа Фибоначчи $O(\log k \cdot n^2)$

$$A_k = A_0^k \qquad A_0 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad A_n = \begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix}$$

Генерация простых чисел

Тестирование на простоту

Вход: N, $|N| = n$

Выход: true, если N - простое

- "В лоб" ("brute force", "решето")

  Перебираем числа от 2 до $\sqrt{N}$
  и проверим, что они не делят N
  сложность $\Theta(n^2 \cdot \sqrt{N}) = \Theta(n^2 \cdot 2^{n/2})$

Малая теорема Ферма

$\exists p$ - простое, $НОД(a, p) = 1 \Rightarrow$

$a^{p-1} \equiv 1 \mod p$

Th. Эйлера

$НОД(a, b) = 1 \Rightarrow a^{\varphi(b)} \equiv 1 \mod b$

Тест Ферма

Возьмём случайное $a \in [2 \dots N-1]$

Если $a^{N-1} \not\equiv 1 \mod N \Rightarrow$

N - составное

Утв: $\exists N$ - составное, $\exists b$: $b^{N-1} \not\equiv 1 \mod N$

$НОД(b, N) = 1$

свидетель непростоты

$\Rightarrow$ Вероятность ошибки теста Ферма $\le \frac{1}{2}$

$\exists a^{N-1} \equiv 1 \mod N$

"плохое" число, тест Ферма ошибается

$c = a \cdot b$ $\quad c^{N-1} = a^{N-1} \cdot b^{N-1} \not\equiv 1 \mod N$

$a^{N-1} \equiv 1$, $b^{N-1} \not\equiv 1$

$\forall$ a - плохого $\exists$ $c = a \cdot b$ - "хороший",
т.е. свидетель непростоты
$\Rightarrow$ Доля плохих $\leq \frac{1}{2}$.

$\exists$ a_1 и a_2 - плохие $\Rightarrow$

$c_1 = a_1 \cdot b$
$c_2 = a_2 \cdot b$ $\Big\rangle$ разные

$\forall$ $c_1 = c_2 \Rightarrow a_1 \cdot b = a_2 \cdot b$

$b = \underbrace{a_1 \cdot a_2^{-1}}_{=1} \cdot b$ // важно, что a_2 обратим $\lhd$

$\Rightarrow a_1 = a_2$

Следствие:

Если мы проверим k случайных чисел от $2 \dots N-1$ $\Rightarrow$ вероятность того, что составное число пройдёт тест Ферма $\leq \frac{1}{2^k}$

Числа Кармайкла

Составные числа без свидетелей непростоты, т.е.

$\forall a \in N \quad a^{N-1} \equiv 1 \mod N$

N - составное, $НОД(a, N) = 1$

Первое такое число $561 = 3 \cdot 11 \cdot 17$

Тест Рабина - Миллера

Берём случайное $a \in [2 .. N-1]$

1. Проверяем, что $a^{N-1} \equiv 1 \mod N$
2. $N-1$ - чётное

$N-1 = q \cdot 2^d$, q - нечётное

$$\Leftarrow \{a^q, \underset{s}{a^{2\cdot q}}, a^{4\cdot q} \dots \underset{1}{a^{2^d\cdot q}}\}$$

Если все эти числа $\pm 1 \Rightarrow$
возвращаем true

Если среди этих чисел есть
число отличное от ± 1, то
найдём последнее такое:

$$\begin{cases} s^2 = 1 \\ s \notin \{-1, 1\} \end{cases} \Rightarrow s \text{ - нетривиальный корень из } 1$$

$\Rightarrow N$ - составное.

! По теореме М.-Р. этот тест
ошибается с вероятностью $\leq \frac{1}{4}$

Следствие

k случайных тестов М.-Р. ошибаются
с вероятностью $\leq 1/4^k$

Как сгенерировать простое?

Возьмём случайное и проверим.

$\pi(N)$ - кол-во простых чисел $\leq N$

$$\pi(N) \sim \frac{N}{\log N}$$

$\Rightarrow$ Случайное число данной n является
простым с вероятностью $\frac{1}{n}$

$\Rightarrow$ В среднем нам нужно $O(n)$
попыток

Замечания:

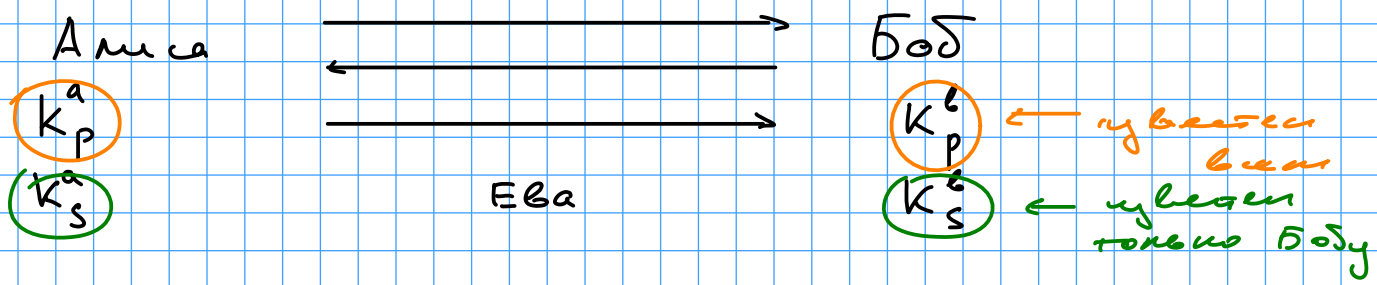
1. Если проверить все числа от 1 до 10^9
с $a = 2$, то получится 0.002% ошибок
(и индустриальные простые числа)

2. ∃ детерминированный алгоритм проверки на простоту $O(n^{12})$
При некоторых предположениях $O(n^4)$
2002

3. Проверка на простоту не даёт разложение на множители

сложная задача

RSA



Алиса шифрует сообщение ключом K_p^b
Боб расшифровывает ключом K_s^b

$C = E(M, K_p^b)$ – сообщение Алисы
$M = D(C, K_s^b)$

Предположение: зная $E(M, K_p^b)$ и K_p^b невозможно эффективно восстановить M

1. ∃ p, q – простые
2. $N = p \cdot q$
3. Возьмём e : $НОД(e, \varphi(N)) = НОД(e, (p-1)(q-1)) = 1$ (секретный ключ)
4. Возьмём d: $d \cdot e \equiv 1 \bmod \varphi(N)$ (из ЕСС(I))

$\equiv E(m, (e, N)) = m^e \bmod N$

(e, N) – публичный ключ

$\equiv D(C, d) = C^d \bmod N = m^{e \cdot d} \bmod N =$

$e \cdot d \equiv 1 \bmod \varphi(n)$

$$= m^{k \cdot \varphi(n) + 1} \bmod N = m^{k \varphi(n)} \cdot m \bmod N = m$$

$$m^{k \varphi(n)} \to 1$$