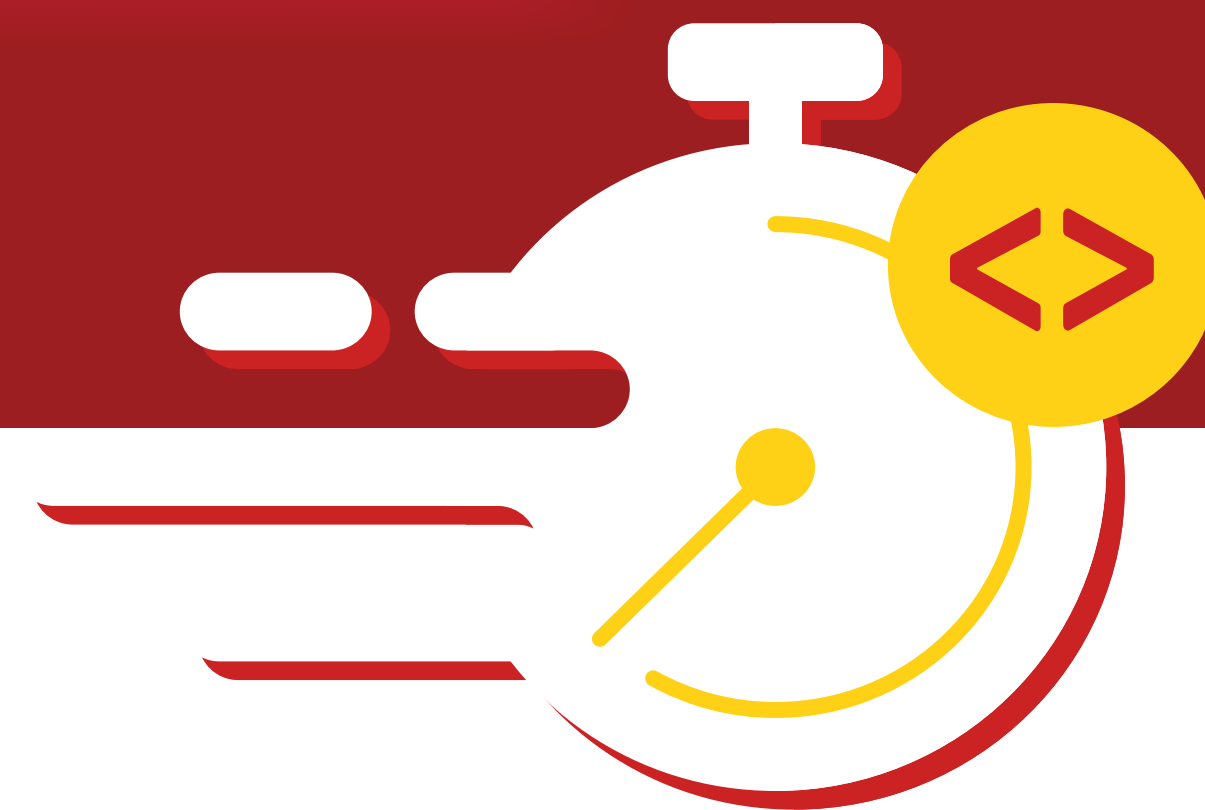
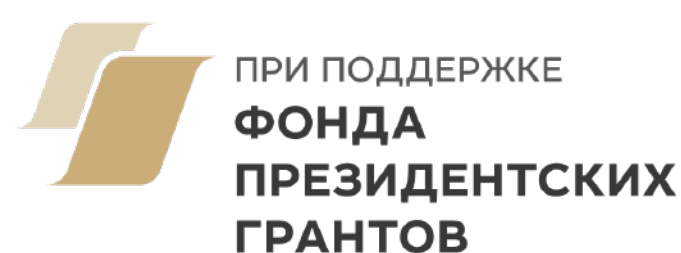


Малая теорема Ферма, бинарное возведение в степень, деление по простому и составному модулю

Урок 2.4



На этом уроке_

- Возведение в степень по модулю
- Деление по модулю
- Деление по простому и по составному модулю

Возведение в степень по модулю: актуальность_

- При умножении по модулю MOD результат не превосходит MOD-1
- Значит, задача «возведите в большую степень по модулю» решается в стандартных типах данных
- Но если показатель степени превосходит 10^9 , соответствующее количество умножений не укладывается по времени
- Надо оптимизировать

Загадка про количество умножений_

- За сколько умножений можно возвести x в 8 степень?

Обычное возведение в степень:

$$x^8 = x * x * x * x * x * x * x * x — 8 \text{ умножений}$$

- А если быстрее?

Можно за 3:

$$x^2 = x * x, \text{ затем } x^4 = x^2 * x^2, \text{ затем } x^8 = x^4 * x^4$$

- А если в 10 степень?

Можно за 4:

$$x^2 = x * x, \text{ затем } x^4 = x^2 * x^2, \text{ затем } x^5 = x^4 * x \text{ и } x^{10} = x^5 * x^5$$

Быстрое возведение в степень: описание_

0. $f(0) = 1$

1. Если $n = 2k + 1$, то $f(n) = a * f(n - 1)$

2. Если $n = 2k$, то $f(n) = f(n / 2) * f(n / 2)$

А какая сложность?

- После шага 1 всегда следует шаг 2 (переход к чётному аргументу), аргумент уменьшается не менее чем вдвое за любые два подряд идущих шага — сложность $O(\log(n))$

Быстрое возведение в степень: код

```
1 long long fastpow (long long a, long long n, long long MOD)
2 {
3     if ( n == 0LL ) return 1LL;
4     if ( n % 2 == 1 ) // нечётное
5         return ( a * fastpow (a, n-1, MOD) ) % MOD;
6     long long tmp = fastpow (a, n/2, MOD); // чётное - сначала считаем an/2
7     return (tmp * tmp) % MOD; // затем возводим в квадрат по модулю MOD.
8 }
```

Деление по модулю: определение_

Обычное умножение и деление: $x = a/b$ обозначает, что $b * x = a$

В случае действий по модулю M :

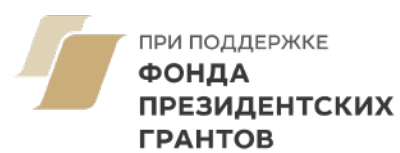
- Разделить a на b — найти такое $0 \leq x < M$, что умножение b на x по модулю M даёт тот же остаток, что и a , то есть $bx = a \pmod{M}$
- Так как остатки периодичны, a , b и x можно заменить остатками от их деления на M , то есть можно считать, что a и b — тоже остатки

Деление по простому модулю: свойства

- Докажем, что при простом P и $(b, P) = 1$ существует такой x , что $bx \equiv a \pmod{P}$
- Лемма: если P — простое, $(k, P) = 1$, $x \not\equiv y \pmod{P}$, то $kx \not\equiv ky \pmod{P}$.

Пусть это не так, тогда $kx - ky = k(x - y)$ делится на P . Но k не делится на P по условию леммы $0 < |x - y| < P$ и тоже не делится на P . Противоречие.

- Рассмотрим остатки от деления на P чисел $b \cdot 0, b \cdot 1, \dots, b \cdot (P-1)$. Согласно лемме, все они попарно различны. Всего различных остатков P , значит, это полный набор остатков. Включающий и тот, что получится от деления a на P



Малая теорема Ферма_

Пусть $Z = 1 * 2 * \dots * (P-1)$.

Так как $1, 2 \dots P-1$ не делятся на P , то $(Z, P) = 1$

- $P-1$ ненулевой остаток
- Любые два числа вида bx , где x – разные ненулевые остатки, различны по модулю P
- bx также пробегает $P-1$ ненулевое значение
- Произведение даёт такой же остаток при делении на P , как и Z

$(b^{P-1} - 1)Z - Z$ делится на P , $Z(b^{P-1} - 1)$ делится на P , $(Z, P) = 1$,
значит: **$b^{P-1} \equiv 1 \pmod{P}$**

Деление по простому модулю_

- Согласно Малой Теореме Ферма, b^{p-1} имеет остаток 1 при делении на P , или же $b^{p-2} * b$ имеет остаток 1 при делении на P
- Но тогда $b^{p-2} \bmod P$ — результат деления 1 на b по модулю P , то есть $1/b \equiv b^{p-2} \pmod{P}$. Домножим на a и получим, что $a/b \equiv a * b^{p-2} \pmod{P}$

Деление по простому модулю: код

Мы используем уже
реализованные ранее
функции умножения
по модулю и возведения
в степень. Параметр модуля
(PMOD) обязан быть простым

```
1 long long Mdiv (long long a, long long b, long long PMOD)
2 {
3     return Mmul (a, fastpow (b, PMOD-2LL, PMOD), PMOD);
4 }
```

Обратное по произвольному модулю_

- Запишем $bx \equiv a \pmod{M}$ в виде $bx + My = a$.
Это диофантово уравнение относительно x и y
- Если a не делится на (M, b) , решений нет
- Иначе запустим расширенный алгоритм Евклида, находим x_0 такое, что $bx_0 + My_0 = (M, b)$,
затем умножим x_0 на $a/(M, b)$ по модулю M

Обратное по произвольному модулю: код

Расширенный алгоритм Евклида должен быть реализован для long long

```
1 long long Mdiv2 (long long a, long long b, long long MOD)
2 {
3     long long x1,x2,y1,y2;
4     long long d=extgcd (b,MOD,x1,y1,x2,y2); // расширенный алгоритм Евклида
5     if (a%d != 0) return -1LL; // возвращаем -1 --- решения нет.
6     return Mmul ((a/d),x1,MOD); // x1 может быть отрицательно, так что не %
7 }
```

Подведем итог_

- Научились выполнять все арифметические действия по модулю
- Вычислили НОД
- Изучили расширенный алгоритм Евклида