**Диофантовы уравнения** — это класс уравнений, которые изучаются в теории чисел и названы в честь древнегреческого математика Диофанта Александрийского. Эти уравнения представляют собой полиномиальные уравнения с целыми коэффициентами, и их решения также должны быть целыми числами. Диофантовы уравнения могут быть линейными или нелинейными, а также иметь одну или несколько переменных.

### Основные особенности диофантовых уравнений:

1. **Целочисленные решения**: Решения таких уравнений ищутся в множестве целых чисел (иногда в множестве натуральных чисел или рациональных чисел).
2. **Полиномиальная форма**: Уравнения имеют вид многочленов от одной или нескольких переменных.
3. **Историческая значимость**: Диофантовы уравнения изучаются с древних времён, и их решение часто связано с глубокими теоретико-числовыми методами.

### Примеры диофантовых уравнений:

1. **Линейные диофантовы уравнения**:  
   Уравнение вида ax+by=c, где a,b,c — целые числа, а x,y — неизвестные, которые нужно найти в целых числах.
   * Например, уравнение 2x+3y=5 имеет бесконечно много решений в целых числах: x=1,y=1; x=4,y=−1; и т.д.
2. **Уравнение Пифагора**:  
   Уравнение вида x2+y2=z2, которое описывает пифагоровы тройки (например, 32+42=52).
3. **Великая теорема Ферма**:  
   Уравнение xn+yn=zn для n>2 не имеет нетривиальных решений в целых числах. Это утверждение было доказано Эндрю Уайлсом в 1994 году.
4. **Уравнение Пелля**:  
   Уравнение вида x2−Dy2=1, где D — целое положительное число, не являющееся полным квадратом. Такие уравнения имеют бесконечно много решений.

### Методы решения диофантовых уравнений:

1. **Метод перебора**: Для простых уравнений можно искать решения путём перебора возможных значений переменных.
2. **Использование алгоритма Евклида**: Для линейных диофантовых уравнений ax+by=c можно использовать алгоритм Евклида для нахождения решений.
3. **Метод факторизации**: Для нелинейных уравнений часто применяется разложение на множители.
4. **Метод бесконечного спуска**: Используется для доказательства отсутствия решений (например, в теореме Ферма).
5. **Использование модульной арифметики**: Анализ уравнения по модулю некоторого числа может помочь в поиске решений или доказательстве их отсутствия.

**Арифметика и свойства сравнений** — это важная часть теории чисел, которая изучает свойства целых чисел и их отношений по модулю. Сравнения позволяют работать с остатками от деления чисел и находят широкое применение в криптографии, алгебре и других областях математики.

### Основные понятия

**Сравнение по модулю**:  
Говорят, что два целых числа a и b сравнимы по модулю m (где m — натуральное число), если они дают одинаковые остатки при делении на m. Это записывается как:

a≡b(mod m).

Это эквивалентно тому, что a−b делится на m (т.е. m ∣ (a−b)).

### Свойства сравнений

1. **Рефлексивность**:

a≡a (mod m).

1. **Симметричность**:  
   Если a≡b (mod m), то b≡a (mod m).
2. **Транзитивность**:  
   Если a≡b(mod m) и b≡c(mod m), то a≡c(mod m).
3. **Сложение и вычитание**:  
   Если a≡b(mod m) и c≡d(mod m), то:

a+c≡b+d(mod m),a−c≡b−d(mod m).

1. **Умножение**:  
   Если a≡b(mod m) и c≡d(mod m), то:

a⋅c≡b⋅d(mod m).

1. **Возведение в степень**:  
   Если a≡b(mod m), то для любого натурального k:

ak≡bk(mod m).

1. **Деление**:  
   Если a≡b(mod m) и d — общий делитель a,b,m, то:

a/d​≡b/d(mod m/d​).

### Линейные сравнения

Линейное сравнение — это сравнение вида:

ax≡b(mod m),

где a,b,m — целые числа, m>0, а x — неизвестная переменная.

#### Решение линейных сравнений

1. **Условие разрешимости**:  
   Сравнение ax≡b(mod m) имеет решение тогда и только тогда, когда gcd(a,m) делит b.
2. **Метод решения**:
   * Если gcd(a,m)=1, то решение единственно по модулю m.
   * Если gcd(a,m)=d>1 и d∣b, то сравнение имеет d решений по модулю m.
3. **Алгоритм решения**:
   * Привести сравнение к виду ax≡b(mod m).
   * Найти d=gcd(a,m).
   * Проверить, что d∣b. Если нет, решений нет.
   * Разделить все коэффициенты и модуль на d:

(a/d)​x≡b/d​(mod m/d​).

* + Решить полученное сравнение с помощью алгоритма Евклида или обратного элемента.

1. **Обратный элемент**:  
   Если gcd(a,m)=1, то существует обратный элемент a−1 по модулю m, такой что:

a⋅a−1≡1(mod m).

Тогда решение сравнения ax≡b(mod m) имеет вид:

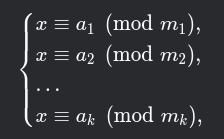
x≡b⋅a−1(mod m).

### Примеры

1. **Пример 1**:  
   Решить сравнение 3x≡4(mod7).
   * gcd(3,7)=1, поэтому решение существует и единственно.
   * Находим обратный элемент для 3 по модулю 7: 3⋅5=15≡1(mod 7), значит, 3−1≡5(mod7).
   * Решение: x≡4⋅5(mod7), то есть x≡20(mod7) или x≡6(mod7).
2. **Пример 2**:  
   Решить сравнение 6x≡9(mod15).
   * gcd(6,15)=3, и 3 делит 9, поэтому есть 3 решения.
   * Делим на 3: 2x≡3(mod5).
   * Решаем 2x≡3(mod5). Обратный элемент для 2 по модулю 5: 2⋅3=6≡1(mod5), значит, 2−1≡3(mod5).
   * Решение: x≡3⋅3(mod5), то есть x≡9(mod5) или x≡4(mod5).
   * Все решения по модулю 15: x≡4,9,14(mod15).

**Китайская теорема об остатках (КТО)** — это важный результат теории чисел, который позволяет решать системы сравнений с взаимно простыми модулями. Она находит применение в криптографии, компьютерной алгебре и других областях. Для практического решения систем сравнений используются алгоритмы Гаусса и Гарнера.

**Формулировка**:  
Пусть даны попарно взаимно простые натуральные числа m1​,m2​,…,mk​ (то есть gcd(mi​,mj​)=1 для всех i≠j) и целые числа a1​,a2​,…,ak​. Тогда система сравнений:



имеет единственное решение по модулю M=m1m2..mk

### Алгоритм Гаусса

Алгоритм Гаусса — это метод решения системы сравнений, основанный на последовательном нахождении решений для каждого уравнения.

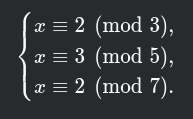
#### Шаги алгоритма:

1. Вычислить M=m1​m2​…mk​.
2. Для каждого i вычислить Mi​=M/mi.
3. Найти обратный элемент Ni​ для Mi​ по модулю mi​, то есть Mi​Ni​≡1(mod mi​).
4. Решение системы:

x≡a1​M1​N1​+a2​M2​N2​+⋯+ak​Mk​Nk​(mod M).

#### Пример:

Решить систему:



* M=3⋅5⋅7=105.
* M1​=35, M2​=21, M3​=15.
* Находим обратные элементы:
* 35≡2(mod3), 2⋅2=4≡1(mod3), значит, N1​=2.
* 21≡1(mod5), 1⋅1=1(mod5), значит, N2​=1.
* 15≡1(mod7), 1⋅1=1(mod7), значит, N3​=1.
* Решение:

x≡2⋅35⋅2+3⋅21⋅1+2⋅15⋅1(mod 105).

x≡140+63+30(mod 105).

x≡233(mod 105).

x≡23(mod 105).

### Алгоритм Гарнера

Алгоритм Гарнера — это более эффективный метод решения системы сравнений, который не требует вычисления обратных элементов для всех модулей. Он особенно полезен, когда модули большие.

#### Шаги алгоритма:

1. Представить модули в виде m1​,m2​,…,mk​, где mi​ попарно взаимно просты.
2. Инициализировать x=a1​.
3. Для каждого i от 2 до k:
   * Вычислить x≡ai​(mod mi​).
   * Найти коэффициент ci​, такой что:

x+ci​⋅(m1​m2​…mi−1​)≡ai​(mod mi​).

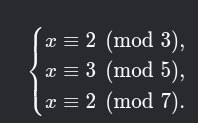
* + Обновить x:

x=x+ci​⋅(m1​m2​…mi−1​).

1. Результат: x — решение системы по модулю M=m1​m2​…mk​.

#### Пример:

Решить систему:



1. Инициализируем x=2.
2. Для второго уравнения:
   * x≡2 (mod 3), 2≡2(mod3).
   * Находим c2​: 2+c2​⋅3≡3(mod5).

2+3c2​≡3(mod5)⟹3c2​≡1(mod5).

Обратный элемент для 3 по модулю 5: 3⋅2=6≡1(mod5), значит, c2​=2.

* + Обновляем x:

x=2+2⋅3=8.

1. Для третьего уравнения:
   * x≡8(mod15), 8≡8(mod15).
   * Находим c3​: 8+c3​⋅15≡2(mod7).

8+15c3​≡2(mod7)⟹1+c3​≡2(mod7).

c3​≡1(mod7).

* + Обновляем x:

x=8+1⋅15=23.

1. Решение: x≡23 (mod 105).

### Сравнение алгоритмов

* **Алгоритм Гаусса** проще для понимания, но требует вычисления обратных элементов для всех модулей.
* **Алгоритм Гарнера** более эффективен, особенно для больших модулей, так как не требует вычисления всех обратных элементов.

**Система остаточных классов (RNS, Residue Number System)** — это система представления чисел, основанная на использовании остатков от деления на набор взаимно простых чисел (модулей). RNS широко применяется в вычислительной технике, криптографии и цифровой обработке сигналов благодаря своей способности выполнять параллельные вычисления и высокой скорости выполнения арифметических операций.

### Основные понятия RNS

1. **Модули RNS**:  
   Выбирается набор взаимно простых чисел m1​,m2​,…,mk​ (то есть gcd(mi​,mj​)=1 для всех i≠j). Эти числа называются модулями RNS.
2. **Диапазон RNS**:  
   Диапазон системы RNS определяется как произведение всех модулей:

M=m1​⋅m2​⋅⋯⋅mk​.

Любое целое число x из диапазона 0≤x<M может быть однозначно представлено в RNS.

1. **Представление числа в RNS**:  
   Число x представляется в виде набора остатков (вычетов) по каждому модулю:

x→(x1​,x2​,…,xk​),

где xi​=x mod mi​.

### Преимущества RNS

1. **Параллелизм**:  
   Арифметические операции (сложение, вычитание, умножение) выполняются независимо по каждому модулю, что позволяет эффективно использовать параллельные вычисления.
2. **Отсутствие переносов**:  
   В RNS нет необходимости учитывать переносы между разрядами, что упрощает выполнение операций.
3. **Высокая скорость**:  
   Операции в RNS выполняются быстрее, чем в традиционных позиционных системах счисления (например, в двоичной системе).
4. **Устойчивость к ошибкам**:  
   RNS позволяет обнаруживать и исправлять ошибки в вычислениях благодаря избыточности модулей.

### Арифметические операции в RNS

1. **Сложение**:  
   Если x→(x1​,x2​,…,xk​) и y→(y1​,y2​,…,yk​), то:

x+y→((x1​+y1​)mod m1​,(x2​+y2​)mod m2​,…,(xk​+yk​)mod mk​).

1. **Вычитание**:

x−y→((x1​−y1​)mod m1​,(x2​−y2​)mod m2​,…,(xk​−yk​)mod mk​).

1. **Умножение**:

x⋅y→((x1​⋅y1​)mod m1​,(x2​⋅y2​)mod m2​,…,(xk​⋅yk​)mod mk​).

1. **Деление**:  
   Деление в RNS выполняется сложнее, так как требует вычисления обратных элементов по каждому модулю.

### **Пример RNS**

Пусть модули RNS: m1​=3, m2​=5, m3​=7. Тогда диапазон RNS:

M=3⋅5⋅7=105.

1. **Представление числа**:  
   Число x=17 представляется в RNS как:

17→(17mod3,17mod5,17mod7)=(2,2,3).

1. **Сложение**:  
   Пусть x=17→(2,2,3) и y=23→(2,3,2). Тогда:

x+y=40→((2+2)mod3,(2+3)mod5,(3+2)mod7)=(1,0,5).

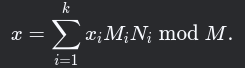
1. **Умножение**:

x⋅y=391→((2⋅2)mod3,(2⋅3)mod5,(3⋅2)mod7)=(1,1,6).

### Преобразование из RNS в позиционную систему

Для преобразования числа из RNS в позиционную систему (например, десятичную) используется **китайская теорема об остатках (КТО)**. Алгоритм преобразования:

1. Вычислить M=m1​⋅m2​⋅⋯⋅mk​.
2. Для каждого модуля mi​ вычислить Mi​=M/mi​.
3. Найти обратный элемент Ni​ для Mi​ по модулю mi​, то есть Mi​Ni​≡1(modmi​).
4. Вычислить число x по формуле:



#### Пример:

Преобразуем (2,2,3) обратно в десятичную систему.

1. M=105, M1​=35, M2​=21, M3​=15.
2. Обратные элементы:
   * 35≡2(mod3), 2⋅2=4≡1(mod3), N1​=2.
   * 21≡1(mod5), 1⋅1=1(mod5), N2​=1.
   * 15≡1(mod7), 1⋅1=1(mod7), N3​=1.
3. Вычисляем:

x=(2⋅35⋅2)+(2⋅21⋅1)+(3⋅15⋅1)mod105.x=140+42+45mod105=227mod105=17.