

# Лабораторная работа №8

Целочисленная арифметика многократной точности

**Студент:** Ван Яо

**Группа:** НФИМд-01-25

**РУДН, 2025**

## Цель работы

1. Изучение теоретических основ арифметических операций с целыми числами многократной точности.
2. Реализация алгоритмов сложения, вычитания, умножения и деления больших целых чисел.
3. Проведение тестирования алгоритмов на различных примерах и анализ их эффективности.

## Основные понятия

- **Большие целые числа:** числа, превышающие стандартные типы данных компьютера
- **b-ичная система счисления:** представление числа в виде  $u = u_1 u_2 \dots u_n$  по основанию  $b \geq 2$
- **Множественная точность:** арифметические операции с числами произвольной длины
- **Алгоритмы:** детерминированные процедуры для выполнения арифметических операций

## Алгоритм 1: Сложение

**Вход:**  $u = u_1u_2\dots u_n$ ,  $v = v_1v_2\dots v_n$ , основание  $b$  **Выход:**

$w = w_0w_1\dots w_n$  (с переносом)

**Псевдокод:** 1.  $j := n$ ,  $k := 0$  2.  $w_j := (u_j + v_j + k) \bmod b$  3.  
 $k := \lfloor (u_j + v_j + k)/b \rfloor$  4.  $j := j - 1$ , если  $j > 0 \rightarrow$  шаг 2 5.  
 $w_0 := k$

**Пример:**  $999 + 1 = 1000$  (десятичная система)

## Алгоритм 2: Вычитание

**Вход:**  $u = u_1u_2\dots u_n$ ,  $v = v_1v_2\dots v_n$ , основание  $b$  **Выход:**

$$w = w_1w_2\dots w_n = u - v$$

**Псевдокод:** 1.  $j := n$ ,  $k := 0$  ( $k$  — заём) 2.  $w_j := (u_j - v_j + k) \bmod b$  3.  $k := \lfloor (u_j - v_j + k)/b \rfloor$  4.  $j := j - 1$ , если  $j > 0 \rightarrow$  шаг 2  
5.  $w_0 := k$

**Пример:**  $1000 - 1 = 999$  (десятичная система)

### Алгоритм 3: Умножение столбиком

**Вход:**  $u = u_1 u_2 \dots u_n$ ,  $v = v_1 v_2 \dots v_n$ , основание  $b$  **Выход:**

$w = w_1 w_2 \dots w_{m+n}$

**Особенности:** - Классический метод “школьного” умножения -

Сложность:  $O(n \cdot m)$  - Использует вложенные циклы для обработки разрядов

**Пример:**  $123 \times 456 = 56088$

## Алгоритм 4 Умножение столбиком

**Вход:**  $u = u_1 u_2 \dots u_n$ ,  $v = v_1 v_2 \dots v_n$ , основание  $b$  **Выход:**

$$w = w_1 w_2 \dots w_{m+n}$$

**Особенности:** - Суммирование произведений для каждого  $s = i + j$  - Более эффективная организация вычислений - Снижение количества промежуточных операций

**Формула:**  $w_{m+n-s} = \sum_{i+j=s} u_{n-i} \cdot v_{m-s+i} \bmod b$

## Алгоритм 5: Деление

**Вход:**  $u = u_n \dots u_1 u_0$ ,  $v = v_t \dots v_1 v_0$ ,  $n \geq t \geq 1$ ,  $v_t \neq 0$  **Выход:**

Частное  $q = q_{n-t} \dots q_0$ , остаток  $r = r_t \dots r_0$

**Ключевые шаги:** - Нормализация чисел - Поразрядное деление с коррекцией частного - Проверка и исправление отрицательных остатков - Формирование окончательного результата



## Анализ и сравнение

			Особенности
Алгоритм	Сложность	Применение	
Сложение	$O(n)$	Базовая операция	Линейная зависимость
Вычитание	$O(n)$	Базовая операция	Требуется $u \geq v$
Умножение	$O(n \cdot m)$	Стандартное умножение	Про-

## Заключение

1. Теоретические знания: Изучены алгоритмы арифметических операций с числами многократной точности.
2. Практические навыки: Реализованы все пять алгоритмов на языке Python с поддержкой различных систем счисления.
3. Результаты тестирования: Все алгоритмы прошли тестирование и показали корректную работу на различных примерах.
4. Значимость работы: Полученные навыки являются фундаментальными для криптографии, компьютерной алгебры и научных вычислений.

Спасибо за внимание!