Лабораторная работа №8

Wang Yao

RUDN University, Moscow, Russian Federation

3 января 2026

Цель работы

Изучить теоретические основы арифметических операций с целыми числами многократной точности, реализовать алгоритмы сложения, вычитания, умножения и деления больших целых чисел, провести тестирование и анализ их эффективности.

Ход лабораторной работы

Основные понятия

• **Большие целые числа**: числа, превышающие стандартные типы данных компьютера

• **b-ичная система счисления**: представление числа в виде $u = u_1 u_2 ... u_n$ по основанию $b \geq 2$

• **Многократная точность**: арифметические операции с числами произвольной длины

• **Алгоритмы**: детерминированные процедуры для выполнения арифметических операций

• **Криптографическое значение**: операции с большими числами лежат в основе современных криптосистем

Алгоритм сложения

Вход: $u = u_1u_2...u_n$, $v = v_1v_2...v_n$, основание b

Выход: $w = w_0w_1...w_n$ (с переносом)

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

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

Сложность: $O(n)$

Алгоритм вычитания

Вход: $u = u_1u_2...u_n$, $v = v_1v_2...v_n$, основание b

Выход: $w = w_1w_2...w_n = u - v$

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

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

Сложность: $O(n)$

Требование: $u \geq v$

Алгоритм умножения столбиком

Вход: $u = u_1u_2...u_n$, $v = v_1v_2...v_m$, основание b

Выход: $w = w_1w_2...w_{m+n}$

Принцип работы: - Классический метод “школьного” умножения - Каждая цифра u_i умножается на каждую цифру v_j - Результаты суммируются с учётом позиций

Пример: $123 \times 456 = 56088$

Сложность: $O(n \cdot m)$

Алгоритм быстрого умножения

Вход: $u = u_1 u_2 ... u_n$, $v = v_1 v_2 ... v_m$, основание b

Выход: $w = w_1 w_2 ... w_{m+n}$

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

Формула:

$$w_{m+n-s} = \sum_{i+j=s} u_{n-i} \cdot v_{m-s+i} \mod b$$

Сложность: $O(n \cdot m)$ (лучшая константа)

Алгоритм деления

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

Выход: Частное $q = q_{n-t} ... q_0$, остаток $r = r_t ... r_0$

Ключевые шаги: 1. Нормализация чисел 2. Поразрядное деление с коррекцией частного 3. Проверка и исправление отрицательных остатков 4. Формирование окончательного результата

Пример: $12345 \div 67 = 184$ (остаток 17)

Сложность: $O(n^2)$

Сравнительный анализ алгоритмов

Алгоритм	Сложность	Память	Применение	Особенности
Сложение	$O(n)$	$O(1)$	Базовая операция	Линейная зависимость
Вычитание	$O(n)$	$O(1)$	Базовая операция	Требует $u \geq v$
Умножение столбиком	$O(n \cdot m)$	$O(n+m)$	Стандартное умножение	Простая реализация
Быстрое умножение	$O(n \cdot m)$	$O(n+m)$	Оптимизированное умножение	Эффективная организация
Деление	$O(n^2)$	$O(n)$	Самая сложная операция	Требует коррекции

Криптографические применения

- **RSA**: умножение и возведение в степень больших чисел
- **Эллиптические кривые**: операции с точками требуют арифметики больших чисел
- **Diffie-Hellman**: модульные операции с большими простыми числами
- **DSA/ECDSA**: генерация и проверка подписей
- **Оптимизация**: эффективные алгоритмы критичны для производительности криптосистем

Ограничения и оптимизации

Ограничения: - Линейные алгоритмы неэффективны для очень больших чисел - Память растёт пропорционально длине чисел - Деление остаётся самой сложной операцией

Оптимизации: - Использование быстрого преобразования Фурье (FFT) для умножения - Блочные алгоритмы для работы с памятью - Параллельные вычисления для ускорения операций - Аппаратная поддержка (специальные процессоры)

Вывод

Изучили и реализовали основные алгоритмы целочисленной арифметики многократной точности.

Основные выводы:

• **Теоретическая ценность**: Понимание фундаментальных алгоритмов компьютерной арифметики

• **Практическая значимость**: Эти алгоритмы являются основой для криптографических систем и научных вычислений

• **Реализованные алгоритмы**: - Сложение и вычитание: линейные алгоритмы $O(n)$ - Умножение: квадратичные алгоритмы $O(n \cdot m)$ - Деление: наиболее сложная операция $O(n^2)$

• **Криптографическое применение**: Эффективная арифметика больших чисел критически важна для производительности криптосистем

• **Итоговый вывод**: Освоенные алгоритмы обеспечивают базовые навыки для работы с большими числами, которые необходимы в криптографии, компьютерной алгебре и вычислительной математике. Для реальных приложений требуются более оптимизированные алгоритмы (Karatsuba, Toom-Cook, Schönhage-Strassen).

Литература

1. Алгоритмы: построение и анализ. — СПб.: Вильямс, 2017.