Лабораторна работа № 8

Целочисленная арифметика многократной точности

Покрас Илья Михайлович

Содержание

# 1 Цель работы

Реализовать алгоритмы целочисленной арифметики многократной точности

# 2 Задание

* Реализовать алгоритм сложения неотрицательных чисел
* Реализовать алгоритм вычитания неотрицательных чисел
* Реализовать алгоритм умножения неотрицательных чисел
* Реализовать алгоритм “быстрый столбик”
* Реализовать алгоритм деления многоразрядных целых чисел

# 3 Теоретическое введение

* Алгоритм сложения неотрицательных чисел заключается в пошаговом суммировании цифр чисел, начиная с младших разрядов и перенося разряды при необходимости. Этот алгоритм позволяет эффективно выполнять операцию сложения.
* Алгоритм вычитания неотрицательных чисел включает поэтапное вычитание цифр чисел, начиная с младших разрядов и заемом разрядов, если необходимо. Он обеспечивает эффективное выполнение операции вычитания.
* Алгоритм умножения неотрицательных чисел базируется на методе пошагового умножения цифр чисел и последующем сложении результатов. Он предоставляет эффективный способ выполнения операции умножения.
* Алгоритм “быстрый столбик” представляет собой метод многоразрядного умножения чисел с использованием оптимизаций. Этот алгоритм значительно ускоряет выполнение операции умножения.
* Алгоритм деления многоразрядных целых чисел основывается на методе пошагового нахождения частного и остатка при делении чисел. Он позволяет эффективно и точно выполнять операцию деления.

# 4 Выполнение лабораторной работы

## 4.1 Алгоритм сложения неотрицательных чисел

Я реализовал функцию алгоритма сложения неотрицательных чисел. Она принимает два массива цифр u и v, их длину n и основание системы счисления b, складывает числа, представленные массивами u и v, с учетом переносов и возвращает результат в виде нового массива цифр w. (рис. 1).

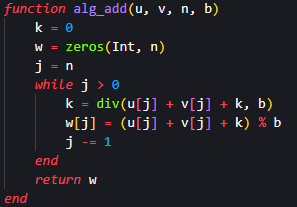


Рис. 1: Алгоритм сложения неотрицательных чисел

## 4.2 Алгоритм вычитания неотрицательных чисел

Я реализовал функцию алгоритма вычитания неотрицательных чисел, принцип которого схож с алгоритмом сложения - она также принимает два массива цифр u и v, их длину n и основание системы счисления b. Она вычитает число, представленное массивом v, из числа, представленного массивом u, с учетом переносов и возвращает результат в виде нового массива цифр w (рис. 2).

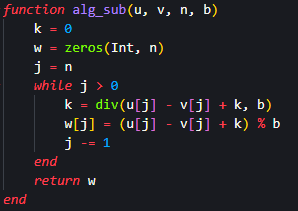


Рис. 2: Алгоритм вычитания неотрицательных чисел

## 4.3 Алгоритм умножения неотрицательных чисел

Далее я реализовал функцию умножения неотрицательных чисел реализует умножение u и v с основанием системы счисления b. В циклах происходит умножение коэффициентов, вычисление остатка от деления и переноса, а также добавление переноса к более старшему разряду. После умножения удаляются ведущие нули и возвращается результат(рис. 3).

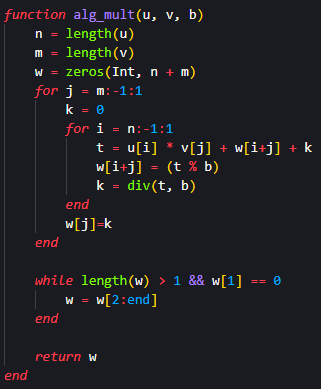


Рис. 3: Алгоритм умножения неотрицательных чисел

## 4.4 Алгоритм “быстрый столбик”

Реализованная мной функция алгоритма “быстрый столбик” также умножает u и v с основанием системы счисления b, но использует более оптимизированный подход. Она выполняет те же операции, но использует вложенные циклы для суммирования произведений коэффициентов во время умножения. Результат также корректируется и возвращается после удаления ведущих нулей (рис. 4).

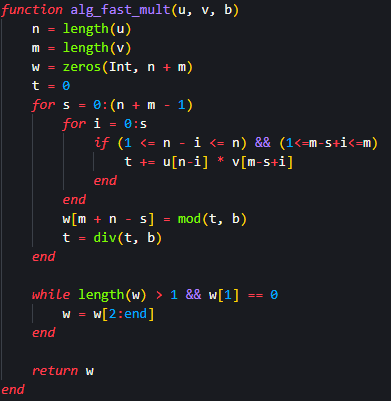


Рис. 4: Алгоритм “быстрый столбик”

## 4.5 Алгоритм деления многоразрядных целых чисел

Сама мною реализованная функция деления многоразрядных целых чисел принимает три параметра: два массива чисел u и v и число b. Внутри функции vec\_convert используется для конвертации массивов чисел u и v в целочисленные значения. Затем в функции alg\_div происходит деление u на v в b-ичной системе счисления. Деление выполняется сначала с помощью оператора ÷, затем находится остаток от деления с помощью оператора %. Далее результат деления d раскладывается на цифры и сохраняется в массив r1, а затем выполняются b итераций, при которых остаток m умножается на 10, далее находится целая часть от деления и добавляется в массив r2. В конце функция возвращает массивы r1 и r2 (рис. 5).

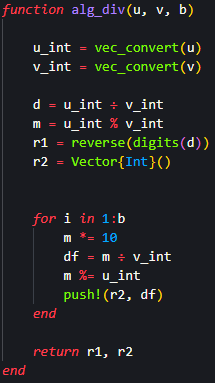


Рис. 5: Алгоритм деления многоразрядных целых чисел

Также Для данного алгоритма была реализована функция конвертации массива в переменную типа BigInt (рис. 6).

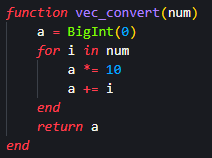


Рис. 6: Функция конвертации массивов

## 4.6 Инициализация переменных и вызов функции

Далее я инициализировал переменные, которые будут входными параметрами вызванных далее функций (рис. 7).

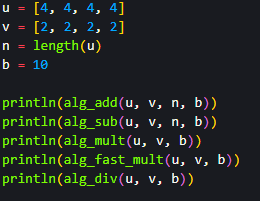


Рис. 7: Инициализация переменных и вызов функции

И получил следующей результат (рис. 8).

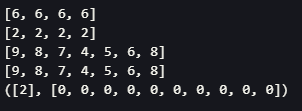


Рис. 8: Результат выполнения кода

# 5 Выводы

Я реализовал алгоритмы целочисленной арифметики многократной точности

# Список Литературы

1. [Julia - Control Flow](https://docs.julialang.org/en/v1/manual/control-flow/)
2. [Julia - Mathematical Operations](https://docs.julialang.org/en/v1/manual/mathematical-operations/)
3. [Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone - Handbook of Applied Cryptography](https://cacr.uwaterloo.ca/hac/)