Отчёт по лабораторной работе №8

Целочисленная арифметика многократной точности

Студент: Гонсалес Ананина Луис Антонио, 1032175329

Группа: НФИмд-02-21

Преподаватель: Кулябов Дмитрий Сергеевич,

д-р.ф.-м.н., проф.

Москва 2021

Содержание

# 1 Цель работы

Цель данной лабораторной работы- изучить теорию и реализовать рассмотренные алгоритмы программно.

# 2 Теоретические сведения

**Целочисленная арифметика многократной точности**

Мы считаем, что числа записаны в *b*-ичной системе счисления, где *b* — фиксированное натуральное число, *b* 2. При этом натуральное число, записываемое не более чем *n* цифрами в *b*-ичной системе счисления, мы обозначаем *u*1 \*. . . u\*\*n\* (допуская, что несколько старших разрядов *u*1, *. . .* , \*u\*\*k\* могут равняться нулю). Основание *b* не всегда равно 2; иногда оно соответствует размеру машинного слова, отведенному под запись обычных целых чисел. В этом случае мы работаем с массивом, содержащим большое целое число[1].

При работе с большими целыми числами удобно хранить знак такого числа в отдельной ячейке или переменной. Если мы хотим, например, перемножить два числа, то знак произведения мы вычисляем отдельно.

**Алгоритм А (сложение неотрицательных целых чисел).**

Для двух неотрицательных чисел *u*1 \*. . . u\*\*n\* и *v*1 \*. . . v\*\*n\* вычисляется их сумма *w*0 \*. . . w\*\*n*; при этом* w\*0 — цифра переноса — всегда равна 0 или 1.

**Алгоритм S (вычитание неотрицательных целых чисел).**

По двум *n*-разрядным неотрицательным целым числам *u*= *u*1 *. . . u****n****v* = *v*1 \*. . . v\*\*n\* 0 вычисляется их разность *w* = *w*1 \*. . . w\*\*n\* = *u − v*.

**Замечание:** Для того, чтобы в общем случае установить, что *u*1 \*. . . u\*\*n\* *v*1 *. . . v****n, надо пройти по цифрам, вычисляя u****j* \*− v\*\*j*. Это простая проверка; с ее помощью находится знак разности* u − v\* в общем случае.

**Алгоритм M (умножение неотрицательных целых чисел столбиком).**

Для чисел *u* = *u*1 \*. . . u\*\*n\* и *v* = *v*1 \*. . . v\*\*m\* в системе счисления с основанием *b* мы находим их произведение *w* = *uv* = *w*1 \*. . . w\*\*m*+*n\*.

**Алгоритм FM («быстрый столбик»)**

| 1 | шаг. *t* := 0. |
| --- | --- |
| 2 | **шаг.** (цикл) Для *s* от 0 до *m* + *n −* 1 с шагом 1 выполнить шаги 3 и |
|  |  |
| 3 | **шаг.** Для *i* от 0 до *s* с шагом 1 выполнить присвоение |
|  | t := t + u*n−i* · v*m−s*+*i*. |
| 4 | **шаг.** Присвоить \*w\*\*m*+*n−s\* := *t* (mod *b*) — наименьший неотрицательный вычет по модулю *b* (опять-таки, это не деление, а чтение записи памяти, если *b* = 2 или *b* — размер машинного слова); *t* := [*t*/*b*]. |

# 3 Выполнение работы



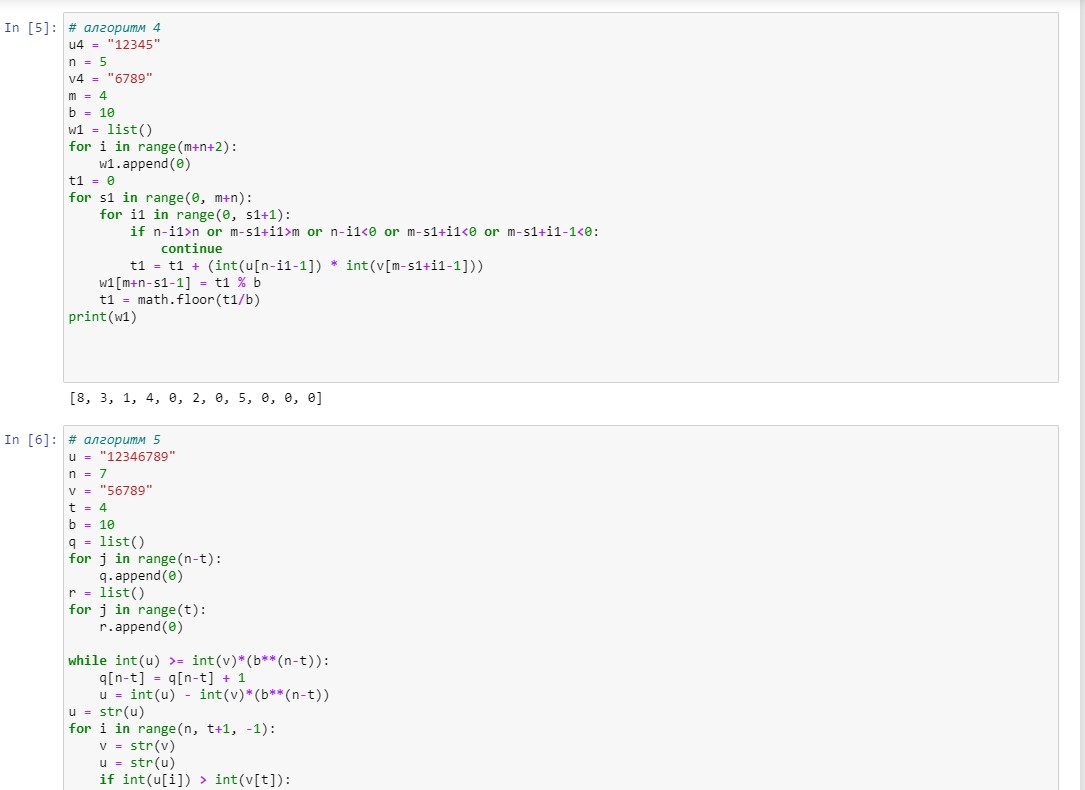
Алгоритм1



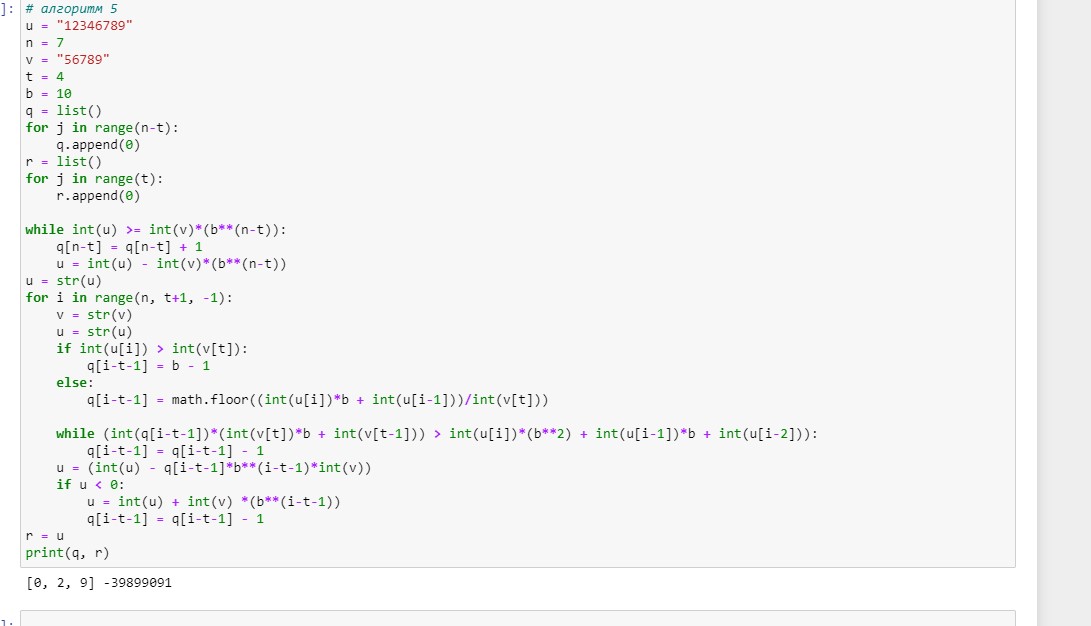
Алгоритм2



Алгоритм3



Алгоритм4



Алгоритм5

# 4 Выводы

В итоге в данной лабораторной работы я изучил теорию и реализовал рассмотренные алгоритмы программно.

# Список литературы

1. Дискретное логарифмирование в конечном поле [Электронный ресурс]. Википедия, 2021. URL: <https://studfile.net/preview/2439346/page:35/>.