МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

УМНОЖЕНИЕ РАЗРЕЖЕННЫХ ПОЛИНОМОВ

ЛАБОРАТОРНАЯ РАБОТА

студента 3 курса 331 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Никитина Арсения Владимировича

Проверил
доцент ______________ А. Н. Гамова

Саратов 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной работе будут рассмотрены принципы алгоритмов вычисления умножения разреженных полиномов, оценки работы алгоритмов в наилучшем и наихудшем случаях, а также программная реализация алгоритмов.

Арифметические операции, такие как умножение, над целыми числами и полиномами лучше изучать в совокупности, так как многие алгоритмы, работающие с целыми числами по существу совпадают с алгоритмами, работающими с полиномами от одной переменной. Это верно не только для таких операций, как умножение и деление, но такое и для более сложно описываемых операций. Например, нахождение вычета целого числа по модулю, задаваемому другим целым числом.

1 Определение полинома

Если i - неотрицательное целое число то размер (i)=$\log i + 1$. Если $p(x)$ - полином, то размер (p)=$CT(p) + 1$, где $CT(p)$ - степень полинома p, то есть наибольшая степень переменной x с нулевым коэффициентом.

Над целыми числами и полиномами моно выполнять приближенное деление. Если a и b - два целых числа и $b \neq 0$, то найдется единственная пара целых чисел q и r, для которых

1. $a = bq + r$
2. $r < b$,

где q и r – соответственно частное и остаток от деления a и b.

Аналогично, если a и b – полиномы, причем b отличен от постоянной, то можно найти такие полиномы, что

1. $a = bq + r$
2. $CT(r) < CT(b)$

2 Разреженные полиномы

Представляя полиномом от одной переменной в виде $\sum_{i=0}^{n-1} a_i x^i$, мы предполагали до сих пор, что он плотный, то есть отличны от нуля почти все его коэффициенты. Для многих приложений полезно предполагать, что полином *разрежен*, то есть число ненулевых коэффициентов много меньше его степени. В такой ситуации логично представлять полином списком пар (a_i, j_i), состоящих из ненулевого коэффициента и соответствующего ему показателя степени переменной x.

3 Умножение разреженных полиномов

Наиболее оптимальная из известных стратегий умножения разреженных полиномов состоит в том, что полином $\sum_{i=1}^{n} a_i x^{j_i}$ задается списком пар:

$$(a_1, j_1), ...(a_n, j_n),$$

где все j_i различны и расположены в порядке убывания, то есть $j_i > j_{i+1}$ для $1 \leq i < n$. Чтобы умножить два полинома, представленных таким образом, находим произведения пар и располагаем их по величине показателей (по вторым компонентам пар), объединяя все члены с одинаковыми показателями. Если это не делать, то придется расплачиваться ростом числа членов с одинаковыми показателями. Поэтому по мере выполнения арифметических операций сложность начнет значительно превосходить ту, которая была бы, если бы приведение подобных членов выполнялась на каждом шаге.

4 Умножение упорядоченных разряженных полиномов

При умножении упорядоченных разряженных полиномов полиномов можно извлечь пользу из их упорядоченности из того, что в одном из них может оказаться много больше членов, чем в другом, чтобы максимально упростить упорядочение множества мономов произведения.

Полиномы $f(x) = \sum\limits_{i=1}^{m} a_i x^{j_i}$ и $g(x) = \sum\limits_{i=1}^{n} b_j x^{k_i}$ списками пар:

$$(a_1, j_1), ..., (a_m, j_m),$$

$$(b_1, k_1), ..., (b_n, k_n),$$

где последовательности $j_1, ..., j_m$ и $k_1, ..., k_n$ монотонно убывают

Полином $\sum\limits_{i=1}^{p} c_i x^{l_i} = f(x)g(x)$, представленный списком пар, в котором последовательность $l_1, ..., l_p$ монотонно убывает.

Предполагаем, что $m \geq n$. Затем:

1. Строим последовательности $S_i, 1 \leq i \leq n$, в которых r-й член, $1 \leq r \leq m$ равен $(a_r b_i, j_r + k_i)$. Таким образом, S_i представляет произведение полинома $f(x)$ на i-й член полинома $g(x)$.
2. Сливаем S_{2i-1} с S_{2i} для $1 \leq i \leq n/2$, приводя подобные члены. Затем попарно сливаем полученные последовательности, приводя подобные члены, и повторяем процесс, пока не получим одну упорядоченную последовательность.

5 Результаты тестирования программы

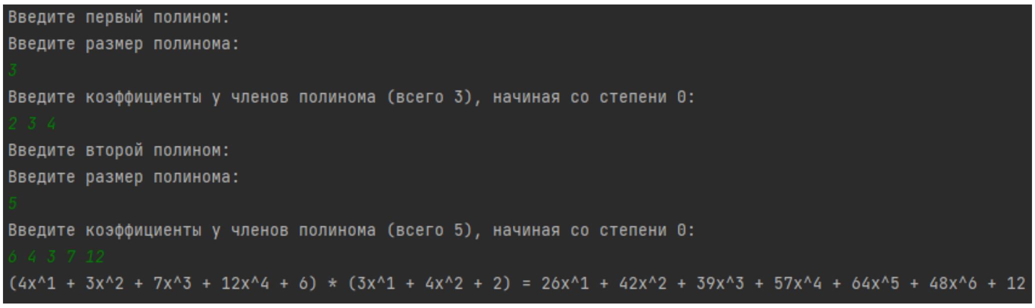

Рисунок 1

6 Программная реализация алгоритма

```python
def polynom_view(coefficients, flag=0):
    polynom = [f'{coefficient}x^{num+1}' for num, coefficient in
    ↪ enumerate(coefficients[1:])]
    polynom.append(coefficients[0])

    if flag == 1:
        print('(', end='')
        print(*polynom, sep=' + ', end=') * ')
    elif flag == 2:
        print('(', end='')
        print(*polynom, sep=' + ', end=') = ')
    else:
        print(*polynom, sep=' + ')
    return


def multiplication(n, a, m, b, res):
    for i in range(n):
        for j in range(m):
            res[i+j] += a[i] * b[j]
    return res


def get_data():
    print('Введите размер полинома:')
    size = int(input())
    print(f'Введите коэффициенты у членов полинома (всего {size}), начиная со
    ↪ степени 0:')
    # polynom = [int(input()) for _ in range(size)]
```

```
    polynom = [int(value) for value in input().split()]
    return size, polynom


print('Введите первый полином:')
n, a = get_data()

print('Введите второй полином:')
m, b = get_data()

if n < m:
    n, m = m, n
    a, b = b, a
res = [0 for _ in range(n + m - 1)]

polynom_view(a, 1)
polynom_view(b, 2)
polynom_view(multiplication(n, a, m, b, res))
```

7 Оценка работы алгоритма

Алгоритм занимает $O(mn \log n)$ времени в предположении, что $m \geq n$:

1. Первая часть алгоритма занимает $O(n, m)$ времени.
2. Второй шаг алгоритма требуется повторить $\log n$ раз, и ясно, что при каждом выполнении процесса вся работа займет $O(mn)$ времени.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены основные определения полиномов и отличия разреженных полиномов от плотных. Также был рассмотрен алгоритм построения умножения разреженных полиномов и оценена скорость его работы. Затем алгоритм был реализован на практике. Перемножение двух разреженных полиномов можно асимптотически оценить в $O(mn \log n)$.