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

АЛГОРИТМЫ АЛГЕБРЫ И ТЕОРИИ ЧИСЕЛ

ЛАБОРАТОРНАЯ РАБОТА №13

студента 4 курса 431 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Никитина Арсения Владимировича

Проверил
доцент _____ А. С. Гераськин

Саратов 2022

СОДЕРЖАНИЕ

1 Задание лабораторной работы

Реализация алгоритма полиномиального деления (PDF).

2 Теоретическая часть

Полиномиальное деление над полем (Polynomial Division over a Field)

Вход: $p_1(x) = \sum_0^m c_i x^i$ и $p_2(x) = \sum_0^n c_i x^i$ над полем f, $n \in \overline{0, m}$ и $d_n \neq 0$. (Этот алгоритм будет работать и над областью целостности J при условии, что d_n обратим в J.)

Выход: $q(x) = \sum_0^{m-n} q_i x^i$ и $r(x) = \sum_0^{n-1} r_i x^i$, обладающие свойством евклидовости.

1. Для k от $m - n$ до 0 выполнять:
   а) $q_k = c_{n+k}/d_n$.
   б) Для j от $n + k - 1$ до 0 выполнять:
      i. $c_j = c_j - q_k d_{j-k}$.
2. Ответ — $q_i,\ i = \overline{0, m - n}$, коэффициенты полинома $q(x)$, вычисленного на шаге 1, и $r_i,\ i = \overline{0, n - 1}$, коэффициенты полинома $r(x)$, где $r_i = c_i$.

3 Практическая часть

3.1 Пример работы алгоритма

```
Выполнить полиномиальное деление PDF - \enter
Выход из программы - 2
Введите значение:
Деление в поле простого числа - 1, деление в поле целых чисел - 2
1
Введите поле, в котором требуется поделить многочлены: 11
Введите коэффициенты полинома, начиная с коэффициента при наибольшей степени:
12 13 14 10 9
Введите от 0 до 5 коэффициентов
Введите коэффициенты полинома, начиная с коэффициента при наибольшей степени:
2 3 4 5
Полином 1 имеет вид:
x^4 + 2*x^3 + 3*x^2 + 10*x^1 + 9*x^0 +
Полином 2 имеет вид
2*x^3 + 3*x^2 + 4*x^1 + 5*x^0 +
(x^4 + 2*x^3 + 3*x^2 + 10*x^1 + 9*x^0 + ) \ (2*x^3 + 3*x^2 + 4*x^1 + 5*x^0 + ) = (2*x^3 + 3*x^2 + 4*x^1 + 5*x^
0 + ) * (6*x^1 + 3*x^0 + ) + (x^1 + 5*x^0 + )

Выполнить полиномиальное деление PDF - \enter
Выход из программы - 2
Введите значение:
Деление в поле простого числа - 1, деление в поле целых чисел - 2
1
Введите поле, в котором требуется поделить многочлены: 2
Введите коэффициенты полинома, начиная с коэффициента при наибольшей степени:
1 0 1
Введите от 0 до 3 коэффициентов
Введите коэффициенты полинома, начиная с коэффициента при наибольшей степени:
1 1
Полином 1 имеет вид:
x^2 + x^0 +
Полином 2 имеет вид
x^1 + x^0 +
(x^2 + x^0 + ) \ (x^1 + x^0 + ) = (x^1 + x^0 + ) * (x^1 + x^0 + ) + ()
```

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
1  import sympy
2
3
4  def polinomial_view(coefs, flag=False):
5      n = len(coefs) - 1
6      a = ''
7      mul = '*'
8      for i, coef in enumerate(coefs):
9          if coef:
10             print(f'{str(coef) + mul if coef != 1 else a}x^{n - i} +',
                   ↪ end=' ')
11     if not flag:
12         print()
13     return
14
```

```
15
16 def get_field():
17     n = int(input('Введите поле, в котором требуется поделить многочлены: '))
18     if not sympy.isprime(n):
19         print('Вы ввели не простое число')
20         return get_field()
21     else:
22         return n
23
24
25 def get_coefs(j=None):
26     print('Введите коэффициенты полинома, начиная с коэффициента при' +
27     ' наибольшей степени:')
28     koef_modula = lambda x : int(x) % j
29     koef_integer = lambda x : int(x)
30     coefs = map(koef_modula if j is not None else koef_integer,
       ↪  input().split())
31     return list(coefs)
32
33
34 def gcdExtended(a, b):
35     if a == 0 :
36         return b, 0, 1
37
38     gcd, x1, y1 = gcdExtended(b % a, a)
39     x = y1 - (b // a) * x1
40     y = x1
41     return gcd, x, y
42
43
44 def divide_in_modula(a, b, j):
45     _, x, _ = gcdExtended(b, j)
46     b_inversed = ((x % j + j) % j)
47     return (a * b_inversed) % j
48
49
50 def pdf(m_coefs, n_coefs, p, flag=False):
51
52     m = len(m_coefs) - 1
53     n = len(n_coefs) - 1
54
```

```
55      right = len(m_coefs) - len(n_coefs)
56      q_coefs = [0] * (right + 1)
57
58      for k in range(right, -1, -1):
59
60          if not flag:
61              q_coefs[k] = divide_in_modula(m_coefs[n + k], n_coefs[n], p)
62          else:
63              q_coefs[k] = m_coefs[n + k] // n_coefs[n]
64
65          for j in range(n + k - 1, k - 1, -1):
66              m_coefs[j] = (m_coefs[j] - q_coefs[k] * n_coefs[j - k])
67              if not flag:
68                  m_coefs[j] %= p
69
70      return q_coefs[::-1], (m_coefs[0:n-1])[::-1]
71
72
73  def main():
74
75      while True:
76
77          print('\nВыполнить полиномиальное деление PDF - \enter')
78          print('Выход из программы - 2')
79
80          try:
81              value = int(input('Введите значение: '))
82
83          except ValueError:
84              value = 1
85
86          if value == 1:
87
88              print('Деление в поле простого числа - 1, деление в поле целых
                ↪  чисел - 2')
89              division_option = int(input())
90              j = None
91
92              if division_option == 1:
93                  j = get_field()
94                  m_coefs_save = m_coefs = get_coefs(j)
```

```
95                print(f'Введите от 0 до {len(m_coefs)} коэффициентов')
96                n_coefs = get_coefs(j)
97            else:
98                m_coefs_save = m_coefs = get_coefs()
99                print(f'Введите от 0 до {len(m_coefs)} коэффициентов')
100               n_coefs = get_coefs()
101
102           print('Полином 1 имеет вид:')
103           polinomial_view(m_coefs)
104
105           print('Полином 2 имеет вид')
106           polinomial_view(n_coefs)
107           if division_option == 1:
108               q, r = pdf(m_coefs[::-1], n_coefs[::-1], j)
109           else:
110               q, r = pdf(m_coefs[::-1], n_coefs[::-1], j, True)
111
112           print('(', end='')
113           polinomial_view(m_coefs_save, flag=True)
114
115           print(')', end='')
116
117           print(' \\ (', end='')
118           polinomial_view(n_coefs, flag=True)
119           print(') = ', end='')
120
121           print('(', end='')
122           polinomial_view(n_coefs, flag=True)
123
124           print(') * ', end='')
125
126           print('(', end='')
127           polinomial_view(q, flag=True)
128           print(') + ', end='')
129
130           print('(', end='')
131           polinomial_view(r, flag=True)
132           print(')')
133
134       elif value == 2:
135           print('Работа программы завершена')
```

```
136                 return
137
138
139 if __name__ == "__main__":
140     main()
```