Отчёт по лабораторной работе №7

Дискретное логарифмирование в конечном поле

Шутенко Виктория Михайловна

Содержание

1 Цель работы **4**

2 Ход работы **5**

2.1 Алгоритм, реализующий p-метод Полларда 5

Список иллюстраций

2.1 Тестирование 9

1 Цель работы

Приобрести практические навыки работы с разложением чисел на множители.

2 Ход работы

2.1 Алгоритм, реализующий p-метод Полларда

```
# Laboratory Work
# Theme: Discrete logarthmification
# Author: Vladimir Doborschuk

# --- Modules ---

import numpy as np

# --- Functions ---

# --- mod(a, b) ---

def mod(a ,b):
    return a % b

# --- find mod order ---

def order(a, p):
    x = 1
    while mod(a**x - 1, p) != 0:
```

```
        x += 1

    return x

# --- Pollard's P-method for Log ---

'''
a - основание
b - значение остатка
p - простое число
'''
def po_method(a: int, b: int, p: int):
    print(f"\n{a}^(x) = {b} mod {p}")
    print("-----------------------------------------------------------------------")
    print('|\tc\t|\tlog c\t|\td\t|\tlog d\t|')
    print("-----------------------------------------------------------------------")

    u = np.random.randint(4)
    v = np.random.randint(4)
    r = order(a, p)

    c = mod(np.power(a, u) * np.power(b, v), p)
    d = c

    u_c, u_d = u, u
    v_c, v_d = v, v

    print(f'|\t{c}\t|\t{u_c}+{v_c}x\t|\t{d}\t|\t{u_d}+{v_d}x\t|')
```

```
def f(x, u_x, v_x):
    if x < r:
        return mod(a*x, p), u_x + 1, v_x
    else:
        return mod(b*x, p), u_x, v_x + 1

c, u_c, v_c = f(c, u_c, v_c)
tmp_d = f(d, u_d, v_d)
d, u_d, v_d = f(tmp_d[0], tmp_d[1], tmp_d[2])

while mod(c, p) != mod(d, p):
    print(f'|\t{c}\t|\t{u_c}+{v_c}x\t|\t{d}\t|\t{u_d}+{v_d}x\t|')
    c, u_c, v_c = f(c, u_c, v_c)
    tmp_d = f(d, u_d, v_d)
    d, u_d, v_d = f(tmp_d[0], tmp_d[1], tmp_d[2])

print(f'|\t{c}\t|\t{u_c}+{v_c}x\t|\t{d}\t|\t{u_d}+{v_d}x\t|')
print("-----------------------------------------------------------------")

x = 1
# print(v_c - v_d, u_d - u_c)
while mod((v_c - v_d)*x, r) != mod(u_d - u_c, r):
    x += 1

print(f"x = {x}")
print(f"\n{a}^({x}) = {b} mod {p}")
print("-----------------------------------------------------------------")
return x
```

```python
# --- Main ---

def main():
    po_method(10, 64, 107)
    po_method(2, 1, 15)


if __name__ == "__main__":
    main()
```

```python
if __name__ == "__main__":
    main()
```

```
10^(x) = 64 mod 107
-------------------------------------------------------------------
|       c       |       log c   |       d       |       log d   |
-------------------------------------------------------------------
|       100     |       2+0x    |       100     |       2+0x    |
|       87      |       2+1x    |       4       |       2+2x    |
|       4       |       2+2x    |       79      |       4+2x    |
|       40      |       3+2x    |       56      |       5+3x    |
|       79      |       4+2x    |       75      |       5+5x    |
|       27      |       4+3x    |       3       |       5+7x    |
|       56      |       5+3x    |       86      |       7+7x    |
|       53      |       5+4x    |       42      |       8+8x    |
|       75      |       5+5x    |       23      |       9+9x    |
|       92      |       5+6x    |       53      |       11+9x   |
|       3       |       5+7x    |       92      |       11+11x  |
|       30      |       6+7x    |       30      |       12+12x  |
-------------------------------------------------------------------
x = 20

10^(20) = 64 mod 107
-------------------------------------------------------------------

2^(x) = 1 mod 15
-------------------------------------------------------------------
|       c       |       log c   |       d       |       log d   |
-------------------------------------------------------------------
|       1       |       0+3x    |       1       |       0+3x    |
|       2       |       1+3x    |       4       |       2+3x    |
|       4       |       2+3x    |       4       |       2+5x    |
-------------------------------------------------------------------
x = 2

2^(2) = 1 mod 15
-------------------------------------------------------------------
```

Рис. 2.1: Тестирование