СПЕЦІАЛЬНІ РОЗДІЛИ

ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №3

Реалізація операцій у скінченних полях характеристики 2

(поліноміальний базис)

1. Мета роботи

Одержання практичних навичок програмної реалізації обчислень у полі Галуахарактеристики 2 в поліноміальному базисі; ознайомлення з прийомами ефективної реалізації критичних по часу ділянок програмного коду та методами оцінки їх ефективності.

Завдання до комп’ютерного практикуму

А) Реалізувати поле Галуа характеристики 2 степеня m в поліноміальному базисі з операціями:

1) знаходження константи 0 – нейтрального елемента по операції «+»;

2) знаходження константи 1 – нейтрального елемента по операції «□»;

3) додавання елементів;

4) множення елементів;

5) обчислення сліду елементу;

6) піднесення елемента поля до квадрату;

7) піднесення елемента поля до довільного степеня (не вище 2m □ 1 , де m –

розмірність розширення);

8) знаходження оберненого елемента за множенням;

9) конвертування (переведення) елемента поля в m -бітний рядок (строкове зображення) і навпаки, де m – розмірність розширення;

Б) Проконтролювати коректність реалізації поля для кожної операції; наприклад, для декількох a, b, c, d перевірити тотожності Додатково можна запропонувати свої тести на коректність. М

В) Визначити середній час виконання операцій у полі. Підрахувати кількість тактів процесора (або інших одиниць виміру часу) на кожну операцію. Результати подати у вигляді таблиць або діаграм.

Хід роботи

Написавши бібліотеку для роботи з елементами в поліноміальному базисі, визначеному моїм варіантом проведемо тести для визначення коректності роботи нашої бібліотеки.

Демонстрація роботи.

```python
#!/usr/bin/python3
# from compmath.poly import *
from compmath.gf import *
from random import getrandbits

def main() -> None:
    fld = GF()
    BITS = fld.m
    A,B = getrandbits(BITS),getrandbits(BITS)
    a,b = fld(A),fld(B)
    print("a,b:", a,b)
    print("poly:", fld.poly)
    print("a+b:",a+b)
    print("a*b:",a*b)
    print("a**2:",a**2)
    a_ = a.inv()
    print("a**-1:",a_)
    print("check:",a*a_)
    f = fld(getrandbits(BITS))
    f.reduce()
    print("f:",f)
    print("a**f:",a**f)
    for i in range(50):
        f = fld(i)
        print(i,f.trace())
if __name__ == "__main__":
    main()
```

```
gratigo@dedsec:~/Documents/term5/SROM/lab3/sources$ ./main.py
a,b: 0x32dccfbfd6510535033361e5658adb8656c4c71ea1b81 0x5f894204d1ad14910900be673802bbd58e18eff000b4f
poly: 0x80000000000000000000000000000000000000000000017
a+b: 0x6d558dbb07fc11a40a33df825d886053d8dc28eea10ce
a*b: 0x62107552b92539fc1cf625b0beabdcdb46831308129ea
a**2: 0x57a0067d6db80bab4d1bdf6b03cb959e15d2d6800c260
a**-1: 0x286363c7b3a431b135b2c5387d8ea777f4969495d3d3e
check: 0x1
f: 0x130c47d4a5474daa4d0af9c31009c5401d63b89af5c15
a**f: 0x9366bb93c0df090ce39397b9ecab72ddf7ad9b2a5915
0 0
1 1
2 0
3 1
4 0
5 1
6 0
7 1
8 0
9 1
10 0
11 1
12 0
13 1
14 0
15 1
16 0
17 1
18 0
19 1
20 0
21 1
22 0
23 1
24 0
```

```
In [1]: from compmath.gf import *

In [2]: from random import getrandbits

In [3]: fld = GF()

In [4]: BITS = fld.m

In [5]: A,B,C = getrandbits(BITS),getrandbits(BITS),getrandbits(BITS)

In [6]: f = fld(2**BITS - 1)

In [7]: a,b,c = fld(A),fld(B),fld(C)

In [8]: a**f
Out[8]: 0x1
```

```
In [11]: (a+b)*c
Out[11]: 0x5e7be330b25b39169439b0be662b7652668cbe701b307

In [12]: b*c + a*c
Out[12]: 0x5e7be330b25b39169439b0be662b7652668cbe701b307

In [13]:
```

Profiler

Add

```
         19 function calls in 0.000 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.000    0.000 <string>:1(<module>)
        1    0.000    0.000    0.000    0.000 gf.py:28(__init__)
        1    0.000    0.000    0.000    0.000 gf.py:46(__add__)
        1    0.000    0.000    0.000    0.000 gf.py:65(bitLen)
        1    0.000    0.000    0.000    0.000 prof_tests.py:10(add)
        1    0.000    0.000    0.000    0.000 {built-in method builtins.exec}
        2    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
        9    0.000    0.000    0.000    0.000 {built-in method builtins.len}
        1    0.000    0.000    0.000    0.000 {built-in method builtins.max}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Mul

```
        2531 function calls in 0.005 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.005    0.005 <string>:1(<module>)
        1    0.000    0.000    0.002    0.002 gf.py:111(reduce)
      164    0.000    0.000    0.001    0.000 gf.py:28(__init__)
       81    0.000    0.000    0.001    0.000 gf.py:46(__add__)
        9    0.003    0.000    0.003    0.000 gf.py:56(mulStep)
      377    0.001    0.000    0.001    0.000 gf.py:65(bitLen)
        1    0.000    0.000    0.005    0.005 gf.py:77(__mul__)
       81    0.000    0.000    0.001    0.000 gf.py:92(lshift)
        1    0.000    0.000    0.005    0.005 prof_tests.py:12(mul)
        1    0.000    0.000    0.005    0.005 {built-in method builtins.exec}
      328    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
     1400    0.000    0.000    0.000    0.000 {built-in method builtins.len}
       82    0.000    0.000    0.000    0.000 {built-in method builtins.max}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
        3    0.000    0.000    0.000    0.000 {method 'pop' of 'list' objects}
```

Pow

```
         498074 function calls (497895 primitive calls) in 0.671 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.671    0.671 <string>:1(<module>)
      268    0.012    0.000    0.417    0.002 gf.py:111(reduce)
    180/1    0.008    0.000    0.671    0.671 gf.py:127(__pow__)
    32375    0.022    0.000    0.155    0.000 gf.py:28(__init__)
    15919    0.045    0.000    0.122    0.000 gf.py:46(__add__)
      801    0.243    0.000    0.243    0.000 gf.py:56(mulStep)
    74744    0.274    0.000    0.279    0.000 gf.py:65(bitLen)
       89    0.001    0.000    0.446    0.005 gf.py:77(__mul__)
    15919    0.043    0.000    0.229    0.000 gf.py:92(lshift)
        1    0.000    0.000    0.671    0.671 prof_tests.py:14(pow)
        1    0.000    0.000    0.671    0.671 {built-in method builtins.exec}
    64751    0.004    0.000    0.004    0.000 {built-in method builtins.isinstance}
   276212    0.014    0.000    0.014    0.000 {built-in method builtins.len}
    16008    0.003    0.000    0.003    0.000 {built-in method builtins.max}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      804    0.000    0.000    0.000    0.000 {method 'pop' of 'list' objects}
```

Inv

```
        18601 function calls in 0.086 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.086    0.086 <string>:1(<module>)
      261    0.000    0.000    0.001    0.000 gf.py:111(reduce)
       87    0.000    0.000    0.005    0.000 gf.py:166(__truediv__)
       88    0.000    0.000    0.000    0.000 gf.py:178(isnull)
        1    0.000    0.000    0.086    0.086 gf.py:183(inv)
     1047    0.001    0.000    0.005    0.000 gf.py:28(__init__)
      433    0.001    0.000    0.003    0.000 gf.py:46(__add__)
     2349    0.074    0.000    0.074    0.000 gf.py:56(mulStep)
     1789    0.005    0.000    0.006    0.000 gf.py:65(bitLen)
      261    0.002    0.000    0.079    0.000 gf.py:77(__mul__)
      172    0.000    0.000    0.002    0.000 gf.py:92(lshift)
        1    0.000    0.000    0.086    0.086 prof_tests.py:16(inv)
        1    0.000    0.000    0.086    0.086 {built-in method builtins.exec}
     2094    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
     8538    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      694    0.000    0.000    0.000    0.000 {built-in method builtins.max}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      783    0.000    0.000    0.000    0.000 {method 'pop' of 'list' objects}
```

Trace

```
        258871 function calls in 0.228 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.228    0.228 <string>:1(<module>)
      180    0.006    0.000    0.216    0.001 gf.py:111(reduce)
      179    0.008    0.000    0.226    0.001 gf.py:127(__pow__)
        1    0.000    0.000    0.227    0.227 gf.py:153(trace)
    16879    0.012    0.000    0.082    0.000 gf.py:28(__init__)
     8349    0.023    0.000    0.064    0.000 gf.py:46(__add__)
    38803    0.144    0.000    0.146    0.000 gf.py:65(bitLen)
     8170    0.022    0.000    0.118    0.000 gf.py:92(lshift)
        1    0.000    0.000    0.227    0.227 prof_tests.py:18(trace)
        1    0.000    0.000    0.228    0.228 {built-in method builtins.exec}
    33758    0.002    0.000    0.002    0.000 {built-in method builtins.isinstance}
   143662    0.007    0.000    0.007    0.000 {built-in method builtins.len}
     8349    0.001    0.000    0.001    0.000 {built-in method builtins.max}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      537    0.000    0.000    0.000    0.000 {method 'pop' of 'list' objects}
```