

Міністерство освіти і науки України

НТУУ«Київський політехнічний інститут»

Фізико-технічний інститут

Спеціальні розділи обчислювальної

Лабораторна робота №4

**Виконав:**

Студент 3 курсу ФТІ

групи ФІ-03

Недождій М. А.

**Перевірила:**

Пекарчук Н. А.

СПЕЦІАЛЬНІ РОЗДІЛИ

ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

Реалізація операцій у скінченних полях характеристики 2 (нормальний базис)

**1. Мета роботи**

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

**3. Завдання до комп’ютерного практикуму**

А) Перевірити умови існування оптимального нормального базису для розширення

(степеня) поля *m* згідно варіанту.

Реалізувати поле Галуа характеристики 2 степеня *m* в нормальному 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 2^m 1, де *m* –

розмірність розширення);

8) знаходження оберненого елемента за множенням;

Код програми виконаний мовою C++, залито на GitHub

#include "NB.h"  
  
int main(){  
 srand((unsigned)time(nullptr));  
 std::deque<int> deque;  
 std::deque<int> deque2;  
 std::deque<int> deque3;  
 for(int i=0; i<MaxLength; i++) {  
 deque.push\_back(rand()%2);  
 deque2.push\_back(rand()%2);  
 deque3.push\_back(rand()%2);  
 }  
  
 std::deque<int> deque4(MaxLength, 1);  
 ONB a(deque);  
 ONB b(deque2);  
 ONB c(deque3);  
 ONB d(deque4);  
  
 show(a.element);  
 show(b.element);  
 show(c.element);  
 std::cout << "\n";  
  
 std::cout << "distibution\n" ;  
 show(a.mul(a.add(a.element, b.element), c.element));  
 show(a.add(a.mul(b.element, c.element), a.mul(c.element, a.element)));  
 std::cout << "\n";  
  
 std::cout << "a\*a^-1=1\n" ;  
 show(a.mul(a.element, a.reverse(a.element)));  
 show(d.element);  
 std::cout << "\n";  
  
 std::cout << "a^2=a\*a\n" ;  
 show(a.square(a.element));  
 show(a.mul(a.element, a.element));  
 std::cout << "\n";  
  
 std::cout << "a^1=a\n" ;  
 std::deque<int> deque5(1, 1);  
 show(a.pow(a.element, deque5));  
 show(a.element);  
 std::cout << "\n";  
  
 std::cout << "a.pow(2)=a.square()\n" ;  
 std::deque<int> deque6 = **{**0, 1**}**;  
 show(a.pow(a.element, deque6));  
 show(a.square(a.element));  
 std::cout << "\n";  
  
 auto startadd = high\_resolution\_clock::now();  
 a.add(a.element, b.element);  
 auto stopadd = high\_resolution\_clock::now();  
 auto durationadd = duration\_cast<microseconds>(stopadd - startadd);  
 std::cout << "time: " << durationadd.count() << " microseconds" << std::endl << std::endl;  
  
 auto startsquare = high\_resolution\_clock::now();  
 a.square(a.element);  
 auto stopsquare = high\_resolution\_clock::now();  
 auto durationsquare = duration\_cast<microseconds>(stopsquare - startsquare);  
 std::cout << "time: " << durationsquare.count() << " microseconds" << std::endl << std::endl;  
  
 auto starttrace = high\_resolution\_clock::now();  
 a.trace(a.element);  
 auto stoptrace = high\_resolution\_clock::now();  
 auto durationtrace = duration\_cast<microseconds>(stoptrace - starttrace);  
 std::cout << "time: " << durationtrace.count() << " microseconds" << std::endl << std::endl;  
  
 auto startmul = high\_resolution\_clock::now();  
 a.mul(a.element, c.element);  
 auto stopmul = high\_resolution\_clock::now();  
 auto durationmul = duration\_cast<microseconds>(stopmul - startmul);  
 std::cout << "time: " << durationmul.count() << " microseconds" << std::endl << std::endl;  
  
 std::deque<int> power = **{**1, 1 ,1 ,1 ,1 ,1 ,1 ,1, 1, 1, 1, 1 ,1 ,1 ,1 ,1 ,1 ,1, 1, 1**}**;  
 auto startpow = high\_resolution\_clock::now();  
 a.pow(a.element, power);  
 auto stoppow = high\_resolution\_clock::now();  
 auto durationpow = duration\_cast<microseconds>(stoppow - startpow);  
 std::cout << "time: " << durationpow.count() << " microseconds" << std::endl << std::endl;  
  
 auto startreverse = high\_resolution\_clock::now();  
 a.reverse(a.element);  
 auto stopreverse = high\_resolution\_clock::now();  
 auto durationreverse = duration\_cast<microseconds>(stopreverse - startreverse);  
 std::cout << "time: " << durationreverse.count() << " microseconds" << std::endl << std::endl;  
  
  
 return 0;  
}

Текст у консолі:

a:

10110000110110101010111100010001010001101100111011000100011111111010010011101110010001001001010101011111011000011011011001100010110001000110110000111110100010111000011110111101010101001000000010011001000110101111011110111010101000111111101000000100001100100110110110001000101111101001010001010001001000000100001100000100000100100101001111100101011001010111011101100100101010001010010010000101111111010110011001001110011

b:

11111011111011111001101101100010001001001000100111100101011110101110011101101010000011101111011010111100110010001001001010110010001110011011001110111001101010001111110001011111010010011110000010101011000001110111100001010111000010010100010101010001111110000010010111110110110010100100011101000001011100110111011001101111111001101000111000101100111110000110110001110000111111100000100001001111100010101000110000010001100

c:

01010010010010101010110011011100111001110001111011011110001000111000100100000101110111011001001111100000101011100110100010010110001110001011000100000011000000000010011110010000001101001110101001101100011100101011101010001111000000010001110011101000010010000000111001010100110100001001101101000100010011001111101000010100000100100111101011000011001000110011011011011010010000011101001001000100100110100010001001000111010

distibution

01110000011111000001000001011001101101101010011101010111111101101100101011110011010111110100110111100101000001000010111101100000000110110010101001100010111010011111110010001001011100110001110011110011010111111001100101101010101110101001110101100011001110100000010010010000000111010011101010111110111001010011001110001010000000000101111100000101101111111000101010011010000010000101001101000001010110010110101111000110011

01110000011111000001000001011001101101101010011101010111111101101100101011110011010111110100110111100101000001000010111101100000000110110010101001100010111010011111110010001001011100110001110011110011010111111001100101101010101110101001110101100011001110100000010010010000000111010011101010111110111001010011001110001010000000000101111100000101101111111000101010011010000010000101001101000001010110010110101111000110011

a\*a^-1=1

11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

a^2=a\*a

11011000011011010101011110001000101000110110011101100010001111111101001001110111001000100100101010101111101100001101101100110001011000100011011000011111010001011100001111011110101010100100000001001100100011010111101111011101010100011111110100000010000110010011011011000100010111110100101000101000100100000010000110000010000010010010100111110010101100101011101110110010010101000101001001000010111111101011001100100111001

11011000011011010101011110001000101000110110011101100010001111111101001001110111001000100100101010101111101100001101101100110001011000100011011000011111010001011100001111011110101010100100000001001100100011010111101111011101010100011111110100000010000110010011011011000100010111110100101000101000100100000010000110000010000010010010100111110010101100101011101110110010010101000101001001000010111111101011001100100111001

a^1=a

10110000110110101010111100010001010001101100111011000100011111111010010011101110010001001001010101011111011000011011011001100010110001000110110000111110100010111000011110111101010101001000000010011001000110101111011110111010101000111111101000000100001100100110110110001000101111101001010001010001001000000100001100000100000100100101001111100101011001010111011101100100101010001010010010000101111111010110011001001110011

10110000110110101010111100010001010001101100111011000100011111111010010011101110010001001001010101011111011000011011011001100010110001000110110000111110100010111000011110111101010101001000000010011001000110101111011110111010101000111111101000000100001100100110110110001000101111101001010001010001001000000100001100000100000100100101001111100101011001010111011101100100101010001010010010000101111111010110011001001110011

a.pow(2)=a.square()

11011000011011010101011110001000101000110110011101100010001111111101001001110111001000100100101010101111101100001101101100110001011000100011011000011111010001011100001111011110101010100100000001001100100011010111101111011101010100011111110100000010000110010011011011000100010111110100101000101000100100000010000110000010000010010010100111110010101100101011101110110010010101000101001001000010111111101011001100100111001

11011000011011010101011110001000101000110110011101100010001111111101001001110111001000100100101010101111101100001101101100110001011000100011011000011111010001011100001111011110101010100100000001001100100011010111101111011101010100011111110100000010000110010011011011000100010111110100101000101000100100000010000110000010000010010010100111110010101100101011101110110010010101000101001001000010111111101011001100100111001

time: 50 microseconds

time: 36 microseconds

time: 17 microseconds

time: 3522288 microseconds

time: 54040429 microseconds

time: 30515743 microseconds

Process finished with exit code 0

**Час виконання операцій у мікросекундах**

|  |  |  |  |
| --- | --- | --- | --- |
| Операція  Значення | Мінімальне | Середнє | Максимальне |
| Додавання | 30 | 68 | 106 |
| Квадрат | 22 | 38 | 54 |
| Слід | 18 | 24 | 30 |
| Множення | 2603094 | 3267245 | 3931396 |
| Піднесення в степінь довжини 20 | 58666079 | 59 166 078 | 59666077 |
| Обернений | 2598310 | 2744347 | 2890384 |