**Алгоритмічні основи криптології**

*Лагун Андрій Едуардович*

**17/9/09 14:23**

**Лекція №1**

# Основні поняття і теореми

## Відображення

Задано дві множини X та Y. Нехай кожному ел-ту мн-ни X поставлено деякий елемент y=f(x) множини Y. В такому випадку задано *відображення* або ф-я f:X->Y із множини X у множину Y.

Відобаження f:X->Y називається ін'єктивним, якщо воно різним аргументам співставляє різні значення:

, 

Відображення з X в Y — сур'єктивне, якщо кожен елемент множини Y має праображ — такий ел-т x, що f(y)=x.

Бієктивним є відображення,яке ін'єктивне і сур'єктивне одночасно.

Тотожне відображення позначається  залишає елементи x на своїх місцях .

Відображення  назив. відображенням лівим оберненим до f:X->Y, за умови, що композиція  є тотожним відображенням, і правим оберненим .

Відображення g називається оберненим до f, якщо воно є одночасно і лівим і правим оберненим до f.

### Властивості оберненого відображення

1. Відображення f:X->Y ін'єктивне тоді і тільки тоді, коли до нього існує ліве обернене.

2. Від. f:X->Y сур'єктивне тоді і тільки тоді, коли до нього існує праве обернене.

3. Якщо відображення f:X->Y бієктивне, то його ліве обернене збігається з правим оберненим.

4. У випадку, якщо множини X та Y — скінчені, і містять однакову к-ть елементів, то відображення f:X->Y має ліве обернене тоді і тільки тоді, коли воно має праве обернене.

### Групи

Групою називається множина G, наділена бінарною операцією \* з такими властивостями:

1.  — асоціативність

2. В G існує нейтральний елемент e такий, що 

3. Для кожного елемента x групи G іс

Якщо підмножина H мн-и G утворює групу, відносно тієї ж операції \*, то вона називається *підгрупою* групи G. Напр., множина раціональних чисел Q утворює групу за додаваням, якій нейтральний елемент 0, а обернений — протилежний, а множина цілих чисел Z є її підгрупою. Множина ++их раціональних чисел  утворює групу за множенням:

e=1, 

Якщо групова операція — множення, то група називається ***мультиплікативною***, її нейтральний елемент є 1-ця, якщо операція ++я, група називається ***адетивною***, нейтральний елемент, а обернений — протилежний елемент.

Для ел-та x групи G =x\*x\*x, де операція \* виконана x-1 раз. В адетивній формі . Для кожного ел-та x скінченої групи для деякого показника m виконується рівність . Найменше з таких m називається порядком ел-та x в групі G. Порядком скінченої групи називається к-ть її елементів. Всі степені ел-та групи утворюють в ній підгрупу, порядок якої = порядку елемента.

Нехай маємо 2 групи G і G' з операціями \* і  відповідно. Відображення f:G->G' зберігає операцію, якщо . Таке відображення називається гомоморфізмом з групи G в групу G'. ***Ядром гомоморфізму*** f:G->G' є мн-а всіх ел-ів G в які f відображає нейтральний ел-т групи G'. Наприклад, відображення, яке кожному цілому числу ставить у відповідність його остачу від ділення наснатуральне n є гомоморфізмом з адетивної групи  в адетивну групу , ядро якого утворює цілі числа кратні n.

Ядро гомоморфізму f:G->G' утворює в G підгрупу. Гоморфізм f:G->G' є ін'єктивним <=>, коли його ядро складається з нейтрального елемента групи G. Таке ядро називається тривіальним.

Гомоморфізм, який є бієктивним відображенням називається ***ізоморфізмом***. Напр., двійковий логарифм  задає ізоморфізм з мультиплікативної групи невід'ємних дійсних чисел  в адетивну групу всіх дійсних чисел .

Для груп G1 і G2 з операція \* і  відповідно,  позначається їх прямий добуток. Множина пар (x1, x2), де  із покомпонентним. Результатом виконання операцій з компонентами x та y множин G вважається ел-т:. Група називається комутативною або абелевою, якщо групова операція має властивість комутативності:

Якщо кожен ел-т групи G є степенем її ел-та g, то цей ел-т називається твірним. Якщо група G має порядок n, то g є її твірним ел-том тоді і тільки тоді, коли його порядок також = n. Група, яка має твірний елемент називається циклічною.

# АЛГОРИТМИ ВИКОНАННЯ ОПЕРАЦІЙ З ДОВГИМИ ЧИСЛАМИ

### Розміщення в пам'яті комп'ютера довгих чисел та аналіз типів даних для виконання арифметичних операцій з ними.

Число **30! = 2 6525 2859 8121 9105 8636 3084 8000 0000** можна представити у вигляді:

++

Можна вважати, що число представлено в 10000-10-вій системі числення, а цифрами числа є 4-значні числа.

Запишемо число у вигляді таблиці

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер ел-ту в масиві A** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Значення** | 9 | 0 | 8000 | 3084 | 8636 | 9105 | 9121 | 2859 | 6525 | 2 |

*Озн.* Числа, для представлення в яких в стандартних комп'ютерних типах даних не вистарчає к-ті двійкових розрядів називаються ***довгими***, а алгоритми реалізації арифметичних операцій з ними — довгою арифметикою.

Алгоритми роботи з довгими числами залежать від представлення цих чисел користувачем в комп'ютері. Довге число можна записати, наприклад, за допомогою масиву десяткових цифр. К-ть ел-тів такого масиву = к-ті значущих цифр в числі. Проте, якщо реалізувати арифметичні операції з цим числом, то розмір масиву має бути достатнім для розміщення результату, наприклад, множення.

**24/9/09 14:23**

**Лекція №2**

продовжуємо

Якщо взяти масив звичайних цілих і вважати його позиційним записом довгого числа в с.ч. з деякою основою В, то кожен елемент масиву набуває значення з діапазону від [0;В-1], а N таких ел-ів дозволять представити числа . Під час представлення довгого числа необхідно виділити місце для запису цього числа. В першу чергу потрібно визначити тип запису довгого числа масиву: з початку чи з кінця масиву, з початку чи з кінця числа.

Нехай N=6, B=10. Розмістимо число 1234. Можливі 4 варіанти:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 |
|  |  | 4 | 3 | 2 | 1 |
| 1 | 2 | 3 | 4 |  |  |
| 4 | 3 | 2 | 1 |  |  |

Ще однією проблемою є заповнення невикористаних розрядів. Під час кодування цих розрядів використовуються такі підходи:

1. кожному довгому числу відповідає змінна цілого типу — *лічильник*, який показує скільки елементів масиву реально використано.

2. невикористані розряди заповнюються значенням, яке не може зустрітися в числі, наприклад "-1".

3. невикористані розряди заповнюються нулями і обробляються так само як і ті, що використовуються.

Останній випадок можливий лише для першого і четвертого варіанту розміщення довгого числа.

### Алгоритми введення/виведення довгих чисел

Позначимо основу системи числення для представлення довгого числа OSN і задамо це значення 10000. Максимальну к-ть цифр довгого числа MAX приймаємо 1000. Алгоритм введення довгого числа розглянемо на прикладі:

Нехай у файлі записано число 51674980124 і в кожному ел-ті масиву А зберігаємо по 4 цифри. Зміну значень ел-ів масиву під час введення зводимо у велику таблицю.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A[0] | A[1] | A[2] | A[3] | c | Коментар |
| 3 | 0124 | 7498 | 516 | - | кінцевий стан |
| 0 | 0 | 0 | 0 | 5 | початковий стан |
| 1 | 5 | 0 | 0 | 1 | 1-й крок |
| 1 | 51 | 0 | 0 | 6 | 2-й крок |
| 1 | 516 | 0 | 0 | 7 | 3-й крок |
| 1 | 5167 | 0 | 0 | 4 | 4-й крок |
| 2 | 1674 | 5 | 0 | 9 | 5-й крок(старша цифра A[1]A[2]) |
| 2 | 6749 | 51 | 0 | 8 | 6-й крок |
| 2 | 7498 | 516 | 0 | 0 | 7-й крок |
| 2 | 4980 | 5167 | 0 | 1 | 8-й крок |
| 3 | 9801 | 1674 | 5 | 2 | 9-й крок |
| 3 | 8012 | 6749 | 51 | 4 | 10-й крок |
| 3 | 0124 | 7498 | 516 | - | 11-й крок |

В A[0] зберігаємо к-ть задіяних (ненульових) елементів масиву A. Під час обробки кожної чергової цифри вхідного числа старша цифра елемента масиву з номером *i* стає молодшою цифрою цифрою числа в елементі *i+1*, а цифра, що вводиться, стає молодшою цифрою числа в A[1]. В результаті роботи алгоритму одержали число, записане ззаду на перед.

*Алгоритм*

1. Обнулюємо елементи масиву A.

2. В A[0] — одиничка

3. Вводиться символ довгого числа

4. Цикл по *i* від старшого розряду до молодшого

4.1. 

4.2. 

5. До молодшого розряду A[1] додається введений символ(c)

6. Якщо в наступному після  розрядів число більше нуля 

7. Перехід до кроку 3.

Усьо.

Для алгоритму виведення необхідно врахувати, що в кожному ел-ті масиву зберігається не послідовність цифр довгого числа, а значення.

Представимо число у вигляді масиву

|  |  |  |  |
| --- | --- | --- | --- |
| A[0] | A[1] | A[2] | A[3] |
| 3 | 3274 | 58 | 1284 |

Під час виведення другого числа з масиву необхідно вивести 0058, інакше буде втрата цифр, тобто незначущі нулі розряду необхідно виводити

*Алгоритм*

1. Виводиться старший розряд масиву A.

2. Цикл по *i* від  по 1.

2.1. Якщо Ai=0, то вивести к-ть нулів, яка дорівнює к-ті цифр основи. Перехід до наступного i.

2.2. Інакше знаходимо к-ть значущих цифр *p*. Поки .

*p=0*

*0123*

*0123/10=012, p=1*

*012/10=01, p=2*

*01/10=0, p=3*

2.3. Поки к-ть цифр основи-p  0, вивести 0. p=p+1.

2.4. Вивести  — значення розряду. Тобто якщо ті всі кроки проскакуються, то відразу виводимо ...

### Алгоритм додавання довгих чисел

Імітуєм додавання стовпчиком, починаючи з молодших розрядів. Для простоти реалізації використовується машинне представлення ззаду на перед. Числа підсумовуються порозрядно. Якщо сума в б.я. розряді виявиться більшою за основу системи числення, то у відповідний розряд результату записується остача від ділення суми на цю основу, а в наступний розряд переноситься одиниця.

Основа 100



Числа, які додаються позначимо b, результат c.

1. Обнулююється масив результату.

2. Знаходимо к-ть розрядів в більшому числі

Якщо 

, інакше 

3. Цикл по i від молодшого розряду до k.

3.1. 

3.2. 

4. Якщо в наступному після  розрядів значення >0, , інакше .

### Реалізація операцій порівняння довгих чисел

Операціями порівняння є: *порівняння на рівність, більше, менше, більше =, менше =*.

Для порівняння використовується нульовий елемент масиву. Якщо к-ть розрядів першого числа більша за к-ть розрядів другого, то підпрограма визначення, наприклад, більшого з чисел, поверне істинне значення.

У випадку нерівності к-ті розрядів відбувається порозрядна перевірка вмісту відповідних розрядів

Алгоритм порівняння на рівність

1. Якщо , числа не рівні. Кінець.

2. Цикл по *i* від молодшого розряду до старшого. Якщо , перехід до наступного i. Інакше числа не рівні. Кінець.

3. Повернути числа рівні.

Алгоритм порівняння більше

1. Якщо , повернути "більше"

2. Цикл по *i* від старшого до молодшого розряду масиву. Якщо , перейти до наступного *i*, якщо , повернути "більше", інакше "не більше".

3. Повернути "не більше"

*Всі інші операції порівняння одержуються за допомогою комбінацій розглянутих алгоритмів*

Алгоритм порівняння із зсувом

Дано A=56784, B=634. B, зсунуте на дві позмиції вліво більше за A. Якщо B=567, то B зсунуте на дві позиції вліво менше за A.







1. Ящо  більше за , повернути 0, інакше:

2. Якщо , повернути 1.

3.  і порівняння на рівність починаючи зі старших розрядів.

3.1. Поки , , 

3.2. Якщо i=Z:

3.2.1. Цикл по i від 1 до Z. Якщо , повернути 0.

3.2.2. Повернути двійку(числа збігаються з врахуванням зсуву). Хріст a=0(залишок).

3.3. Інакше повернути 1.

**10/8/2009 2:23 PM**

**Лекція №4**

**Тема:** продовжуємо.

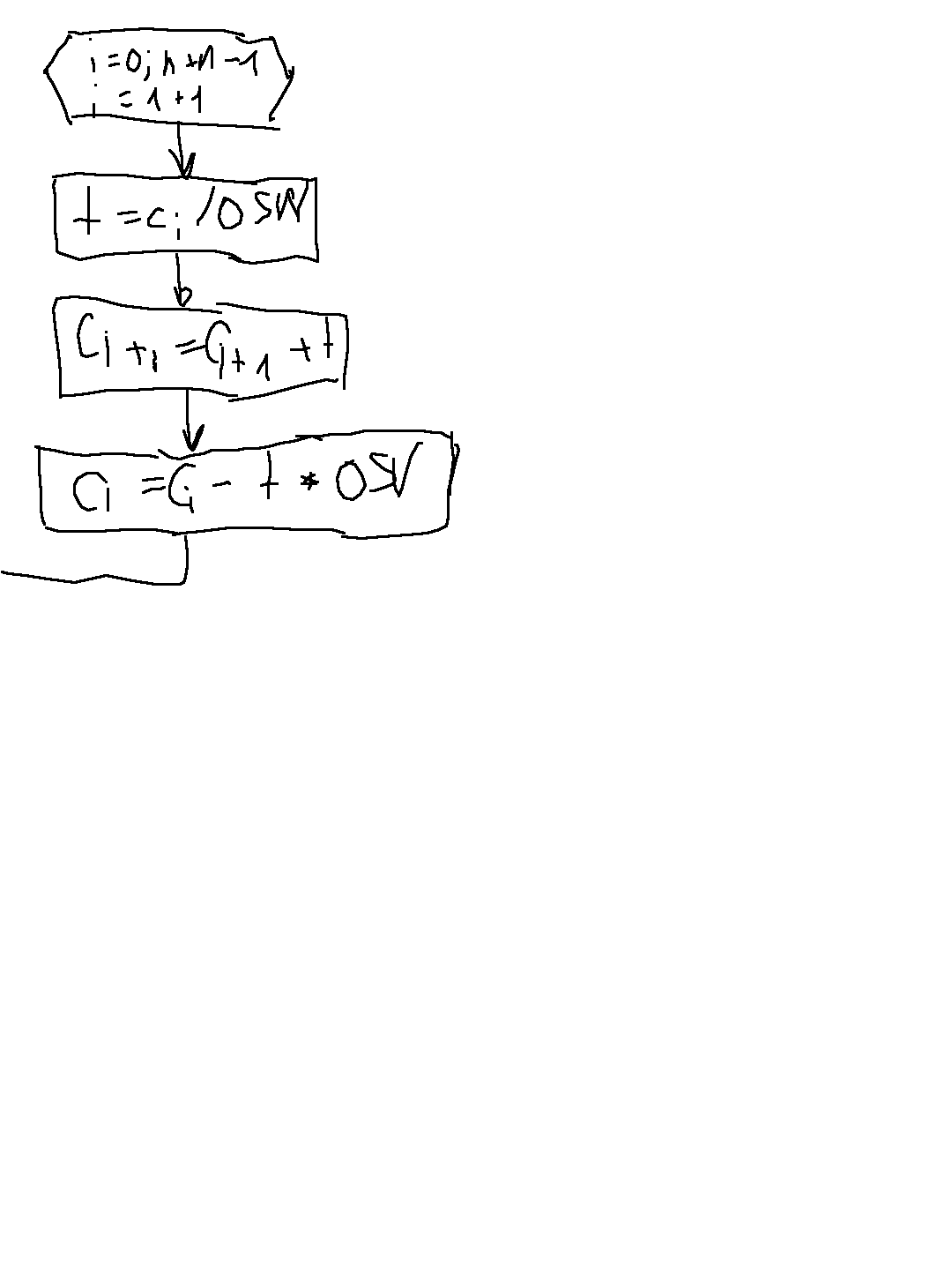
Коефіцієнти згортки дають результат множення многочлена на многочлен , де m I n довільні числа.

*Конструкцію, яку утворюють коефіцієнти Ci називають пірамідою множення*

|  |  |
| --- | --- |
|  |  |

Базовий тип, який використовується для зберігання коефіцієнтів повинен мати достатній обєм порядку . Для множення довгих чисел з використанням згортки Фурє, достатнь:

1. Обчислити коефіцієнти згортки Ci (і=0; n+m-1)
2. Зробити необхідні перенесення, щоб всі коефіцієнти були менші за основу. Блок-схема:



Основна складність алгоритму множення полягає в обчисленні коефіцієнтів згортки на першому кроці, для цього використовується швидке перетворення Фурє (ШПФ), або швидке перетворення Хартлі (ШПХ). ШПФ комплексного вектора обчислюється як комплексний вектор з координатами

Де - комплексний корінь енного степеня з одиниці.

Індекс степеня n м.б. відсутній, тоді степінь кореня = кількості координат вектора, що перетворюється. До швидкого існує обернене, яке обчислюється:

В швидкому використовується теорема про згортку: перетворення Фурє від згортки двох векторів є скалярним добутком Фурє образів цих векторів

Алгоритм обчислення згортки складається з трьох кроків:

1. Швидке перетворення ШПФа,б
2. Скалярно перемножити одержані вектори
3. Обчислити зворотнє перетворення Фурє від скалярного добутку

Перемноження многочленів зводиться до скалярного перемноження відповідних векторів. Вважаємо, що на кожному кроці розміри векторів однакові і = n, оскільки не можна скалярно перемножити короткий вектор на довший. Найбільшої швидкодії досягають варіанти швидкого перетворення Фур’є, що працюють з векторами розміру , тому вектори за необхідності доповнюють нулями. Приклад:

Числа зберігаються з заду наперед, тобто старший розряд – останній. Алгоритм швидкого множення:

1. Знайти найменше число – степінь двійки, такий, що . Для нашого випадку, L=2+5=7=8.
2. Доповнити a і b – ведучими нулями:
3. Обчислити ШПФ дійсних векторів на обох масивах цифр
4. Скалярно перемножити перетворені вектори, одержавши вектор, довжиною *L*.
5. Застосувати зворотне перетворення Фур’є, результатом якого буде згортка.
6. Перетворити згортку в масив цілих чисел. Зробити перенесення.

Цифри для довгих чисел зберігаються в цілочисельному форматі, тому для швидкого перетворення Фур’є їх необхідно скопіювати в тимчасові масиви типу з плаваючою крапкою. Якщо необхідно одержати результат максимальної довжини *n*, то необхідно виділити для довгого числа пам’ять, не меншу за . Наприклад, якщо максимальний результат буде складатися із тисячі цифр, за основою *OSN*, то мінімальний об’єм пам’яті = 1024. Всі обчислення проходять у форматі з плаваючою крапкою, тобто результат буде наближеним, бо кожну координату вектора округлюють до найближчого цілого числа. Це найбільша проблема використання множення з швидким перетворенням Фур’є у криптографії.

# Обмеження ШПФ множення

Нехай потрібно перемножити вектори з n координат з дійсними коефіцієнтами, тоді похибка множення оцінюється зверху виразом:

, де - точність додавання (віднімання), - точність тригонометричних обчислень. З цієї формули можна знайти *мінімально можливе значення основи*. Наприклад, для типу *double* ***(53 біти)*** , похибки тригонометрії обмежені величиною , обмежимо верхню межу похибок , приблизно порахувавши константи, одержимо:

Для чисел довжиною значення основи менше 4100. На практиці якщо вибрати основу 10000, можна використати значно більші довгі числа. Під час округлення необхідно враховувати різницю між точним значенням і результатом округлення. Якщо вона менша за 0.2, то множення як правило дає правильний результат. Інакше рекомендується зменшити основу або скористатися іншим базовим типом для зберігання коефіцієнтів. Після виконання п’ятого кроку поперднього алгоритму маємо тільки згортку – результат без перенесень. Значення коефіцієнтів згортки можуть бути набагато більшими за основу, досягаючи максимального значення , під час зворотного перетворення відбувається ділення результату на *n*. Тобто максимальний розмір цифри рівний . Тому для запобігання переповнення основа системи числення вибирається не більшою ***чотирьох*** десяткових чисел. Існує три проблеми виконання операції множення:

1. Точність тригонометрії;
2. Точність під час обчислення ШПФ;
3. Переповнення базового типу.

Друга і третя проблеми вирішуються шляхом зменшення основи системи числення або збільшення базового типу, при цьому ефективність алгоритму спадає, оскільки менша основа означає збільшення кількості цифр, а більший базовий тип не завжди доступний.

# Використання ШПХ для обчислення згортки

Для коефіцієнтів , теорема про згортку має вигляд:

**(\*)**

C=ДПХ(а), d=ДПХ(b),

Індекси обчислюються за |n|, тобто замість елементу з індексом n, беруть елемент з нульовим індексом. Всі кроки попереднього алгоритму залишаються без змін за виключеням четвертого кроку, де координати вектора обчислюються за формурою \*. В швидкому перетворенні Хартлі існує два стилі:

* ШПФ за частотою – ШПФ\_ЧТ
* ШПФ за часом – ШПФ\_ЧС

Це дозволяє уникнути виклику підпрограми перестановки в алгоритмі ШПФ, яка займає 10% алгоритму множення. ШПФ\_ЧС має *на вході перетворений* вектор, а в кінці роботи *повертає звичайний*. А ШПФ\_ЧТ – навпаки. Кроки попередньо розглянутого алгоритму змінюються так:

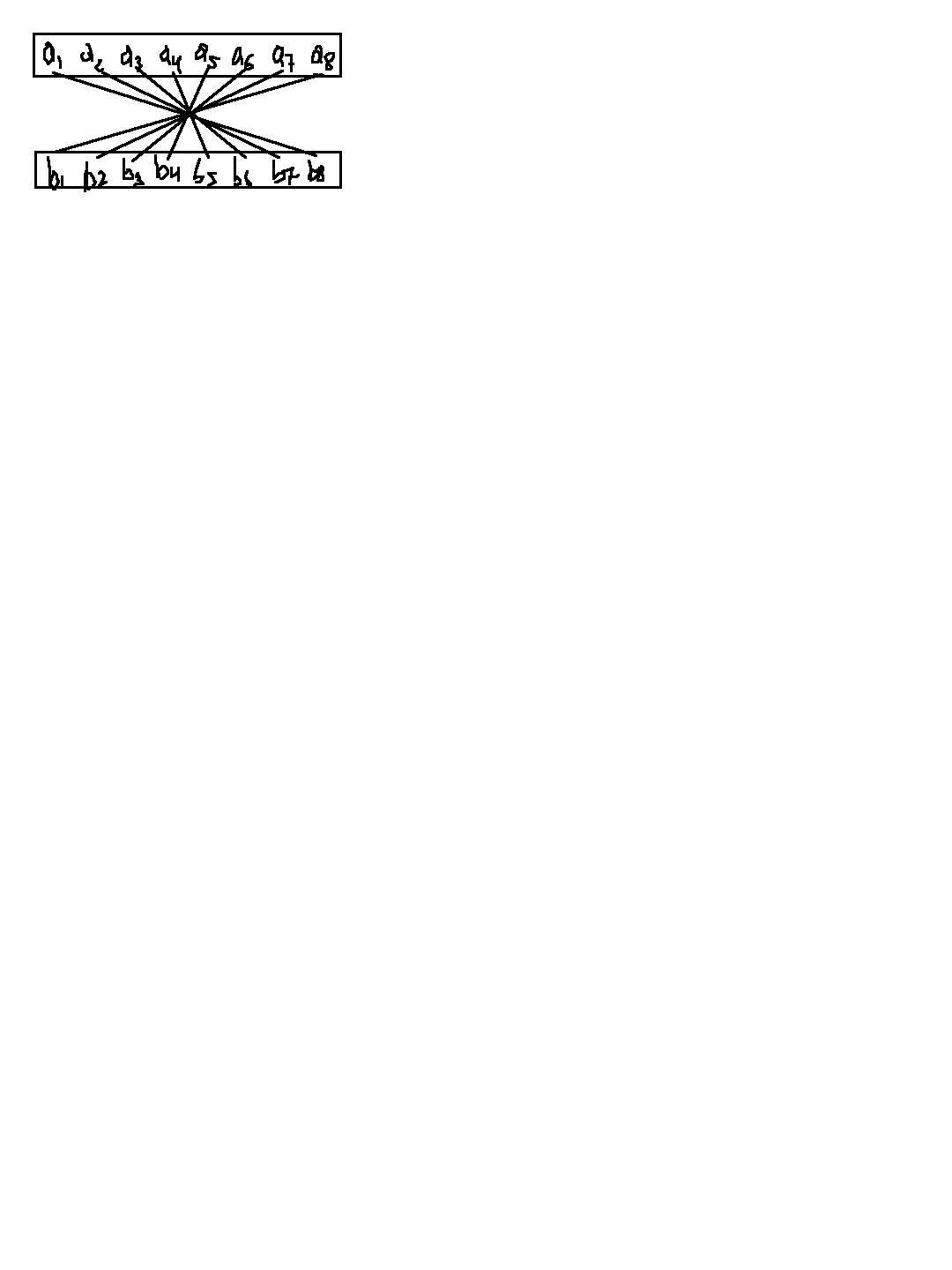
1. Вектори зберігаються із звичайним порядком індексів;
2. Обчислюється ШПХ\_ЧТ для обох масивів;
3. Вектор на виході в перетвореному вигляді;
4. Обчислюються коефіцієнти ШПХ за формулою \*, враховуючи зміни в порядку індексів;
5. Вектор ще у перетвореному вигляді
6. Обчислити ШПХ\_ЧС, результатом якого буде згортка.

Останній вектор має звичайний порядок індексів. Приклад обчислення для векторів:

Елементи з індексом 0 I n/2 обробляються окремо, оскільки для них:

Для інших елементів необхідно врахувати, що на вхід формули числа подаються з двох позицій, а вихід записується в одну. Щоб уникнути затирання можна обчислювати по два значення одночасно використовуючи симетричність виразу:

Елементи вектора а будуть перетворюватись поперно і на одному місці. Утвориться метелик згортки, виконання якого для , дасть необхідний результат. Схема обчислення для 8-елементних векторів:



Обчислення пари коефіцієнтів дискретного перетворення Хартлі потребує чотирьох додавань і чотирьох множень. Для перетворення Фур’є, кількість операцій становить 2 комплексних множень (8 звичайних) і 4 додавання, проте комплексний вектор – в 2 рази коротший, тому дискретне перетворення Хартлі потребує на 2 додавання більше для пари елементів.

[Дата]

Лекція №5

# Порівняльна характеристика алгоритмів множення довгих чисел

Перемножуємо два числа однакового розміру. В залежності часу виконання операції множення від довжини числа для алгоритмів множення стовпчиком і швидке перетворення Фур’є.



Зелене – множення стовпчиком, суцільна – швидке перетворення Фур’є.

Стрибки на основі алгоритму швидкого перетворення Фур’є виникають під час переходу через степінь двійки. Починаючи із 150 розрядних чисел алгоритм на основі швидкого перетворення Фур’є працює значно швидше

# Основи теорії чисел

## Ділення з остачею

Дано 2 цілих числа. Число a ділиться на b, якщо знайдеться таке ціле q, що a=b/q. Синоніми: а кратне b, b – дільник а. Відповідно позначення ,

Нехай – рівність сум цілих чисел. Якщо всі доданки в цій рівності крім одного кратні до b, то той, що залишився мусить бути кратним b.

Теорема. Для даного цілого відмінного від нуля числа b довільне ціле число a представляється у вигляді , .

*Доведення*. Одне представлення числа a рівністю одержимо, якщо взяти рівним найбільшому кратному числа b, що не перевищує a.



Нехай є два представлоення

|  |  |
| --- | --- |
|  | - |

В останній рівності b-q1 ділиться на b, отже r-r1 повинно ділитися на b, оскільки

*,*

*Озн.* Число q називається неповною часткою, а r – залишком від ділення . Залишок завжди невідємне число, а неповна частка може бути довільним цілим числом. Пр. . Залишок 1, неповна частка -2.

## Найбільший спільний дільник та його властивості

*Озн.* Ціле число d, що ділить одночасно числа називається спільним дільником цих чисел. Найбільше d з такою властивістю називається найбільшим спільним дільником цих чисел (НСД). Позначається .

Основні властивості НСД:

1. Якщо , , то
2. Якщо , – НСД.
3. Якщо , то сукупність спільних дільників a i b збігаються із сукупністю спільних дільників b I c, зокрема: ;
4. Нехай , тоді
5. То саме з діленням
6. Випливає з попередніх двох
7. Якщо , тоді

*Озн*. Цілі числа a і b називаються взаємнопростими, якщо їх НСД = 1. Іншими словами, числа a і b взаємнопрості лише тоді, коли знайдуться такі цілі числа, що

# Алгоритм Евкліда

Слово **алгоритм** є основою латинізованого імені арабського математика *«ал-Хорезма», Абу, Абдула, Мухаммед, Ібн, ал-Маджусі* і означає в сучасному сенсі деякі правила, інструкції та команди, виконуючи які можна досягнути необхідного результату. Алгоритм, який дозволяє за заданими натуральними числами a і b знайти їх НСД, називаються **алгоритмом Евкліда**. Дано 2 числа.

, вважаємо

Алгоритм.

1. Ввести a,b
2. Якщо b=0, вивести a, кінець.
3. r=залишок від a,b

1. Перейти до 2.

В сучасному буквенному записі алгоритм Евкліда виглядає так:

, тобто процес обірветься максимум через n кроків. Покажемо, що . Переглянувши рівності зверху вниз бачимо, що довільний дільник а і b ділиться на . Якщо переглянути ланцюжочок в зворотньому порядку, тоді видно що є дільником .

; тобто - НСД.

Сукупність дільників a і b збігаються із сукупністю дільників НСД . Це дає практичний спосіб знаходження чисел і , таких, що . З ланцюжища рівностей виражаємо наші а і b.

Приклад. Знайти НСД

|  |  |
| --- | --- |
|  | 763=242\*3+37  242=37\*6+20  37=20\*1+17  20=17\*1+3  17=3\*5+2  3=2\*1+1  2=1\*2 |

## Лінійні діофантові рівняння

Довільне рівняння (як правило з цілими коефіцієнтами) називаються ***діофантовим***, якщо його потрібно розв’язати в цілих числах.

Дано лінійне діофантове рівняння: , ,

Нехай , тоді

В такого рівняння є розв’язок (пара цілих чисел x і y) тільки тоді, коли (d дільник c). Нехай так воно і є. Поділимо обидві частини рівняння на d і будемо вважати, що a і b взаємнопрості, тобто . Розглянемо 2 випадки.

1. *Нехай c=0*

Тоді початкове рівняння - *однорідне лінійне* діофантове рівняння

*,* оскільки x має бути цілим числом, то , де - довільне ціле число(параметр). Тоді . Тобто розв’язками лінійного однорідного діофантового рівняння є пари чисел вигляду , . Множина всіх таких пар називається ***загальним розв’язком лінійного однорідного діофантового рівняння***, аб.я. конкретна пара з цієї множини – частковим розв’язком.

1. *Нехай*

*Теорема.* Якщо a і b – взаємнопрості, - часткових розв’язок лінійного діофантового рівняння , то його загальний розв’язок має вигляд:

(\*)

*Тобто загальний розвязок лінійного діофантового рівняння є сумою загального розвязку відповідного однорідного рівняння і деякого часткового розвязку неоднорідного рівняння.*

*Доведення*. Те, що праві частини рівностей \* дійсно є розв’язками перевіряється їх безпосередньою підстановкою в початкове рівняння. Нехай деякий загальний розв’язок рівняння , тоді

|  |  |
| --- | --- |
|  | - |

–

Частковий розвязок знаходиться так, оскільки (a,b)=1, знайдуться такі a,b, що , u,v знаходяться з алгоритму евкліда. Помноживши

2009-10-22 14:15

Лекція №6

Тема: Приклад

Останні два рівняння – частковий розв’язок. Загальний розв’язок

# Прості числа

Натуральне число називається простим, якщо воно має лише два додатних дільники – 1 і p. Інші натуральні числа, крім одиниці називаються складеними.

Число одиниця за домовленістю – ні просте, ні складене.

*Спостереження 1:* найменший дільник будь-якого числа , відмінний від 1 є простим числом. Доведення. Нехай , c – найменше з такою властивістю. Якщо існує c1, таке, що , то , .

*Спостереження 2:* Найменший, відмінний від 1 дільник складеного числа , не перевищує . Дано: , c – найменше з такою властивістю.

Для складання таблиці простих чисел існує процедура, яка називається ***решетом Ератосфена***. В чому соль? Записуємо натуральні числа



Йдемо по натуральному ряду зліва направо. Підкреслюємо перше непідкреслене і не викреслене число, а з подальшого ряду викреслюємо всі числа, кратні підкресленому.

**Написати програмку на модулі**

Коли викреслено всі кратні до простих менші за p, то ті, що залишилися не викреслиними і менші за – прості. Це означає, що складання таблиці всіх простих чисел, менших за a закінчено одразу, як тільки викреслені всі кратні до простих, менших за .

# Ланцюгові дробіки

## Розкладання чисел в ланцюжкові дробіки

*Означення*. Ланцюговим (або неперервним) дробом називається вираз, вигляду:

Вважаємо, що - цілі, решта – натуральні. Числа

Називаються прямуючими дробами ланцюгового дробу . Ланцюговий дріб може бути ***скінченним*** (має скінченну кількість дробових ліній і неповних часток) і ***нескінченним*** (вниз і вправо). В першому випадку дріб є певним раціональним числом, а в другому випадку – всі неповні части – раціональні числа. Значенням нескінченного ланцюгового дробу є границя послідовності його прямуючих дробів

**Теорема**. Довільне дійсне число може бути розкладене в ланцюговий дріб єдиним чином, і навпаки – довільний скінченний і нескінченний ланцюговий дріб має значення – дійсне число.

Нехай – число, яке міститься між послідовними цілими числами. a називається нижнім цілим числом,

- нижні і верхні цілі числа

,

Продовжуючи проце взяття нижнього цілого і обертання дробових частин одержимо розклад довільного числа в ланцюговий дріб.

Оскільки , то процес зациклиться і одержимо нескінченний ланцюговий дріб, в якому всі неповні частки, починаючи з другої = 2. Висновок: довільне ірраціональне число, якщо і можливо, то можливо представити у вигляді нескінченного ланцюгового дробу.

Маємо число альфа раціональне у виглядлі а поділити на бе, тобто у вигляді дробу двичайного. За таких умов процес розкладання числа в ланцюговий дріб зажди скінченний і можна використати алгоритм Евкліда.

*,*

*,*

,

…

,

,

- неповні части. Теорему доведено.

**Приклад.**



Обчислення прямуючих дробів

Прямуючий дріб заміною букви виразом . Якщо багатоповерховий прямуючий дріб спростити, одержимо деяке раціональне число вигляду - одноповерховий дріб. - чисельники, – знаменники прямуючого дробу . Для зручності приймаємо

[дописать]

Рекурентні формули для обчислення чисельників і знаменників прямуючих дробів. Ці формули разом з початковими умовами , дозволяють обчислювати довільні прямуючі дроби.

**Приклад**. Обчислити прямуючі дроби ланцюгового дробу

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|  |  | 2 | 6 | 1 | 1 | 1 | 2 | 2 |
|  | 1 | 2 | 13 | 15 | 28 | 43 | 114 | 271 |
|  | 0 | 1 | 6 | 7 | 13 | 20 | 53 | 126 |

**2009-10-29 14:24**

**Тема: Властивості прямуючих дробіків**

1. Для будь-якого s>0, дріб нескоротний.

Нехай НСД , тоді d ділить різницю , яка дорівнює , що неможливо.

*Доведення.* Відомо, що

*В Найповільніше знаменники будуть зростати при , тобто при*

Ця рекурентна формула разом з початковими умовами задає послідовність Фібоначі. Характеристичне рівняння для послідовності Фібоначі таке:

Його корені

*. Звідси*

1. *Для довільного нескінченного л.д. послідновність його прямуючих дробів сходиться*
2. *Нехай альфа…*

*, ніж до*

*Доведення. На ес плюс першому кроці замінюємо , тому маємо точну рівність*

*З рівності випливає, що різниці в дужках – різних знаків. Крім того*

1. Для будь-якого альфа розклад в л.д. – єдиний.

Дов. Нехай маємо два розклади одного числа

Якщо два числа рівні, то в них рівні цілі частини, отже збігаються обернені велечини до дробових … далі за індукцією

# Континуанти. Аналіз алгоритму Евкліда

Дано рекурентні вирази для чисельників і знаменників прямуючих дробів. Розглянемо тридіагональний визначник

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| q1 | 1 | 0 | 0 | … | 0 | 0 |
| -1 | q2 | 1 | 0 | 0 | 0 |
| 0 | -1 | q3 | 1 | 0 | 0 |
|  |  |  | … |  |  |
| 0 | 0 | 0 | 0 | qn-1 | 1 |
| 0 | 0 | 0 | 0 | -1 | qn |

Визначник називається континуантою третього порядку. Числа є неповними частками з алгоритму Евкліда, тому є цілими. Розкладемо континуанту по останньому стовпцю.

Одержаний вираз подібний до рекурентних виразів чисельників і знаменників прямуючих дробів.

**Лема1**. Континуанта дорівнює сумі всіх добутків елементів , один з яких містить всі ці елементи, а інші одержуються з нього відкиданням однієї чи декількох пар множників із сусідніми номерами. Якщо відкинули всі залишилася одиниця.

Твердження Леми справедливе для континуант 2го порядку.



Крок індукції.

Нехай твердження Леми вірне для континуант порядків, тоді континуанта записується:

Добутки, які одержуться від множення на Qn доводять лему. Кількість доданків в континуанті Енного порядку є сумою кількості доданків в континуантах n-1 і n-2 порядків. Тобто континуанта містить складову – n+1 число Фібоначі.

**Лема 2**.

Доведення. База індукції

Крок індукції

Тоді

## Аналіз алгоритму Евкліда. Теорема Ламе.

Нехай задано такі, що алгоритму Евкліда для обробки a і b необхідно виконати n ділень із залишком, при чому а – найменше з такою властивістю. Тоді:

**Доведення**. Розкладемо a/b в ланцюговий дріб

, де – неповні частки з алгоритму Евкліда. За умови теореми їх n штук. За властивістю **3**  прямуючих дробіків континуанти і **взаємнопрості**. Тобто якщо , тоді

|  |  |
| --- | --- |
|  | (\*) |

Для скінченного л.д.

Оскільки континуанта є многочленом від всіх цих змінних мінімальне значення досягається при рівності

Підставляючи ці формули в зірочку