Група ІПС-41, Шатохін Максим Сергійович

Тема: Алгоритм Евкліда. Варіант 17

Завдання.

Знайти d - найбільший спільний дільник чисел a,b,

розв'язати рівняння au+bv=d,

розв'язати рівняння as+bt=c,

Для прикладів 1 і 2 навести всі кроки алгоритму, для прикладу 3 можна навести тільки 5 перших і 5 останніх кроків.

4. Розробити програму і провести чисельний експеримент для дослідження часової складності алгоритму. Навести результати експерименту для прикладів. Навести псевдокод вибраного Вами алгоритму.

5. Навести теоретичний матеріал по обчисленню часової складності для трьохстрічкової машини Тьюрінга.

1. a = 284

b = 160

c = 20

Відповідь:

d = 4

u = -9

v = 16

s = -45

t = 80

2. a = 40033175

b = 3282300

c = 11375

Відповідь:

d = 175

u = 5669

v = -69143

s = 368485

t = -4494295

3. a = 247678868885655

b = 230200504829220

c = 35175

Відповідь:

d = 15

u = 4638525773129

v = -4990713715584

s = 10877342937987505

t = -11703223663044480

**Розв’язання**

|  |  |
| --- | --- |
| **1. Прямий хід** | **Зворотній хід** |
| 284 = 1\*160 + 124 | 124 = 1\*a - 1\*b |
| 160 = 1\*124 + 36 | 36 = –1\*a + 2\*b |
| 124 = 3\*36 + 16 | 16 = 4\*a – 7\*b |
| 36 = 2\*16 + 4 **(d = 4)** | 4 =–9\*a + 16\*b |
| 16 = 4\*4 + 0 | **(u = -9, v = 16)** |
|  |  |

|  |  |
| --- | --- |
| **2. Прямий хід** | **Зворотній хід** |
| 40033175 = 12\*3282300 + 645575 | 645575 = 1\*a + (-12)\*b |
| 3282300 = 5\*645575 + 54425 | 54425 = -5\*a + (61)\*b |
| 645575 = 11\*54425 + 46900 | 46900 = 56\*a + (-683)\*b |
| 54425 = 1\*46900 + 7525 | 7525 = -61\*a + (744)\*b |
| 46900 = 6\*7525 + 1750 | 1750 = 422\*a + (-5147)\*b |
| 7525 = 4\*1750 + 525 | 525 = -1749\*a + (21332)\*b |
| 1750 = 3\*525 + 175 | 175 = 5669\*a + (-69143)\*b |
| 525 = 3\*175 + 0 |  |
|  | **(u = 5669, v = -69143)** |
|  | |

**3. Прямий хід**

247678868885655 = 1\*230200504829220 + 17478364056435

230200504829220 = 13\*17478364056435 + 2981772095565

17478364056435 = 5\*2981772095565 + 2569503578610

2981772095565 = 1\*2569503578610 + 412268516955

2569503578610 = 6\*412268516955 + 95892476880

...

4665 = 1\*4020 + 645

4020 = 6\*645 + 150

645 = 4\*150 + 45

150 = 3\*45 + 15 **(d = 15)**

45 = 3\*15 + 0

**Зворотний хід**

17478364056435 = 1\*a + (-1)\*b

2981772095565 = -13\*a + (14)\*b

2569503578610 = 66\*a + (-71)\*b

412268516955 = -79\*a + (85)\*b

95892476880 = 540\*a + (-581)\*b

...

4020 = 42181120784\*a + (-45383793975)\*b

645 = -50495940777\*a + (54329930789)\*b

150 = 345156765446\*a + (-371363378709)\*b

45 = -1431123002561\*a + (1539783445625)\*b

15 = 4638525773129\*a + (-4990713715584)\*b

**(u = 4638525773129, v = -4990713715584)**

**4.** Проведемо чисельний експеримент використовуючи мову програмування Python 3.8. Будемо заміряти час виконання функції знаходження НСД алгоритмом Евкліда запускаючи по 10000 разів для кожної пари чисел, які будуть отримуватися випадковим чином з певного інтервалу, не є взаємно простими та мають НСД рівний 1.

Тестуємо функцію нсд(a, b).

Усі отримані результати записуються у файл. Їх буде продемонстровано далі на графіку.

Псевдокод з мови Python:

нсд(a, b):

поки a > 0 та b > 0:

якщо a > b:

a := a mod b

інакше:

b := b mod a

повернути a + b

виміряти\_час\_роботи\_функції(початок\_проміжку, кінець\_проміжку,

кількість\_повторних\_запусків):

a := випадкове\_ціле(початок\_проміжку, кінець\_проміжку)

b := випадкове\_ціле(початок\_проміжку, кінець\_проміжку)

якщо a mod b == 0 або b mod a == 0:

b += 1

d := випадкове\_ціле\_зі\_зменшене(початок\_проміжку,

кінець\_проміжку)

a \*= d

b \*= d

start := поточний\_час()

повторити кількість\_повторних\_запусків:

res := нсд(a,b)

end := поточний\_час()

повернути start-end

тестування():

interval = 5

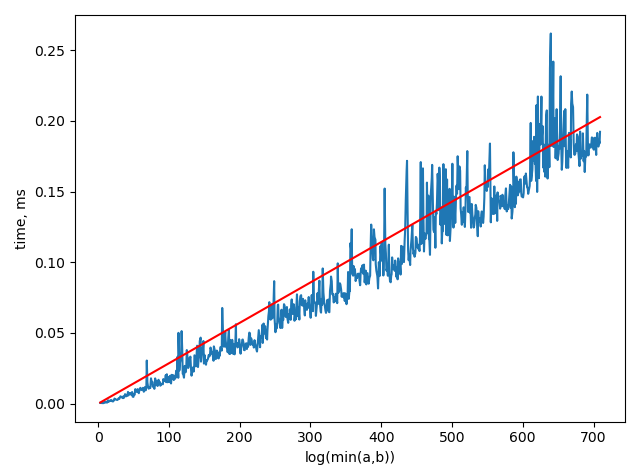
для кожного значення лічильника від 1 до 1000:

записати\_результат(виміряти\_час\_роботи\_функції(

interval, 2\*interval, 10000))

interval \*= 2

Результати відображені на графіку з логарифмічною шкалою синім кольором.



На вертикальній осі відображено час виконання тесту пару в мілісекундах. Графік з помаранчевим кольором – асимптотична оцінка часу виконання алгоритму Евкліда (log(min(a,b))).

**5**. Для трьохстрічкової машини Тьюрінга, яка обчислює часову та просторову складность алгоритму Евкліда, будемо використовувати наступний алфавіт:

{𝐵, 1, 𝑋} – де В означає прогалину, а Х являє собою допоміжний символ.

Для кодування чисел у даному випадку зручно використлвувати унітарний код, де число N буде записуватися з символів «1» у кількості N штук. Таким чином на вхід (перша стрічка) машини Тьюрінга, яка реалізовує алгоритм Евкліда для чисел M та N (де M<N) буде подаватися два набори символів «1» через прогалину.

Після цього данні з вхідної стрічки копіюються на робочу, другу, стрічку і голівка машини Тьюрінга для другої стрічки встановиться на позицію першого символу «1» зліва. Після завершення роботи результат знаходження НСД буде записаний в унітарному кодуванні, на вихідній стрічці шляхом копіювання результату з робочої області.

Опишемо алгоритм Евкліда:

1. a <- N, b <-M

2. t <- |a - b|; якщо t=0 – зупинка

3. a <- t, b <- min{a,b}; goto 2

Розглянемо реалізацію набору операцій алгоритму для машини Тьюрінга.

Віднімання:

Гарантована умова того, що перше число b буде менше або дорівнювати а, тому віднімання ab реалізується затиранням символів «1» з числа а у кількості b.

Також необхідно відслідковувати які символи з числа b були враховані, для цього будемо

помічати їх символом «Х».

S1, 1 -> S2, X, R

S2, 1 -> S2, 1, R

S2, B -> S3, B, R

S3, B -> S3, B, R

S3, 1 -> S4, B, L

S4, 1 -> S4, 1, L

S4, B -> S4, B, L

S4, X -> S1, X, R

S1, B -> S5, B, L

S5, X -> S5, 1, L

S5, B -> END\_SUBTRACT, B, R

Цей процес займає a + b + 1 клітинку та b(b + 2) переходи.

Перевірка чи t = 0:

З алгоритму бачимо, що при t = 0 НСД буде рівним b. При цьому після процесу віднімання необхідно замінити символи «Х» на «1». Для цієї перевірки на нульову різницю необхідно подивитися в наступну клітинку після затирання «1». Ця дія збільшить кількість переходів віднімання до b(b + 3). І призводить до виходу робочої стрічки у режим HALT та виводу на третю стрічку. Після віднімання голівка машина Тьюрінга знаходиться на першій «1» b, необхідно зберегти умову того, що b<a, тобто в даних умовах b<t, це виконаємо занесенням b <- min{a,b} Для цього маємо викреслити з b |t-b| символів, якщо потрібно.

END\_SUBTRACT, 1 -> M1, X, R

N1, 1 -> M1, X, R

M1, 1 -> M1, 1, R

M1, B -> M2, B, R

M2, B -> M2, B, R

M2, 1 -> M3, X, L

M2, X -> M2, X, R

M2, B -> R1, B, L

M3, X -> M3, X, L

M3, B -> M3, B, L

M3, 1 -> M3, 1, L

M3, X -> N1, X, R

N1, B -> R2, B, L

R2, X -> R2, X, L

R2, B -> C1, B, R

R1, 1 -> R1, B, L

R1, B -> R1, B, L

R1, X -> R2, B, L

C1, X -> C1, 1, R

C1, B -> C2, B, R

C2, B -> C2, B, R

C2, X -> C3, 1, R

C3, X -> C3, 1, R

C3, B -> K1, B, L

K1, 1 -> K1, 1, L

K1, B -> K2, B, L

K2, B -> K2, B, L

K2, 1 -> K3, 1, L

K3, 1 -> K3, 1, L

K3, B -> S1, B, R

На цю дію знадобиться b(b + 1) + b + t ~ b(b + 3) переходів та a + b + 1 клітинок.

Загалом для однієї ітерації алгоритму Евкліда знадобиться:

b(b + 3) + b(b + 2 + 1) = b(2b + 6) = 2b(b + 2) переходи.

Отже складність алгоритму Евкліда можна оцінити як 𝑂(𝑏2logb), де b – кількість символів у числі min(M, N).

Просторова складність алгоритму Евкліда складає M + N + 1.