

Бінарний алгоритм піднесення до степені

Курс: ООП в ФВЕ

Київський Національний Університет імені Тараса Шевченка

29 травня 2020 р.

Стандартний алгоритм:

```
19
20 long long int power (int a, int n) {
21     if (n == 0)
22         return 1;
23     if (n == 1)
24         return a;
25     else {
26         long long int b = power (a, n-1);
27         return b * a;
28     }
29 }
30
```

Бінарний алгоритм:

```
6
7 using namespace std::chrono;
8
9 long long int binpow (int a, int n) {
10     if (n == 0)
11         return 1;
12     if (n % 2 == 1)
13         return binpow (a, n-1) * a;
14     else {
15         long long int b = binpow (a, n/2);
16         return b * b;
17     }
18 }
19
```

Нехай F_1 і F_2 - перша пара чисел Фібоначі. Тоді векторно:

$$F_{12} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix},$$

Для отримання пари наступних чисел F_{23} можна побудувати матрицю перетворення:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

Тоді

$$F_{23} = A \cdot F_{12} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} F_2 \\ F_1 + F_2 \end{pmatrix} = \begin{pmatrix} F_2 \\ F_3 \end{pmatrix}$$

І Взагалі,

$$F_{i(i+1)} = A \cdot F_{(i-1)i} = A^n \cdot F_{12} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} F_1 \\ F_2 \end{pmatrix},$$

Для піднесення до степені матриці A ефективніше використовувати бінарний алгоритм.

Нехай дано набір із n точок $\{p_i\}_n$ і m перетворень, які потрібно зробити з цими точками. Кожну операцію потрібно виконати $length$ разів. Операції бувають трьох типів:

- ▶ Зсуву;
- ▶ Масштабування;
- ▶ Оберт навколо вісі.

Необхідно розрахувати координати після перетворень.

Для кожної з операцій складність бінарного алгоритму становитиме:

$$O(m \cdot \log(length))$$

Кількість шляхів фіксованої довжини у графі

Нехай дано граф G (n вершин), число k . Мета - для кожної пари вершин i і j знайти кількість шляхів між ними довжиною k вершин.

Розв'язок:

Будується матриця суміжності графі і підноситься до степені k . Елементи отриманої матриці і будуть розв'язками.

Складність алгоритму становитиме:

$$O(n^3 \cdot \log(k))$$

Інша модифікація даної задачі містить зважений граф. Метою є знайти шлях, який містить рівно k ребер і має найменшу вагу. Розв'язок даної модифікації відрізняється, однак теж містить бінарний алгоритм піднесення матриці до степені.

Варіація бінарного піднесення до степені: добуток двох чисел за модулем

Дано: Числа a , b і m , такі, що поміщаються в стандартний тип даних. При цьому добуток $a \cdot b$ - ні.

Мета: Знайти значення $[a \cdot b \pmod m]$ не використовуючи довгу арифметику.

Розв'язок: Бінарне сумування