

II

Принцип Теллегея (транзюции)

Уже Есть матрица A $n \times m$ и алгоритм $f_A : \mathbb{R}^m \rightarrow \mathbb{R}^n$, который за s операций

знает $v \mapsto Av$

Тогда $\exists$ алгоритм $f_{A^T} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, который за s с операций знает $u \mapsto A^T u$

„Модель“: a_i - ячейки памяти. В них приходит вход и надо получить выход алгоритма:

Также есть „константы“ алгоритма: $n, m, A_{ij}, \sum_{i=0}^{2^n} i \cdot A_{0 \ i \% m}$ - считаем, что уже есть и

считать их не надо

Алгоритм состоит из элементарных операций таких типов:

1° добавить в конец
ячейку с 0

2° удалить предыдущую
ячейку

$$3^\circ a_i * = c$$

$$4^\circ a_i += a_j * c$$

Матрицы, соотв этим операциям:

$$1^\circ \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ 0 & & 1 & \\ & & & \ddots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

$$2^\circ \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ 0 & & 1 & \\ & & & \ddots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

$$3^\circ \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ 0 & & 1 & \\ & & & \ddots \\ 0 & c & \dots & 0 \end{pmatrix}$$

$$4^\circ \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ 0 & & 1 & \\ & & & \ddots \\ 0 & c & \dots & 0 \end{pmatrix}$$

Алгоритм - посл-во операций с матрицами $H_1, \dots, H_c$ т.е. $A = H_c \cdot H_{c-1} \cdot \dots \cdot H_1 \Rightarrow$

$\Rightarrow A^T = H_1^T \cdot \dots \cdot H_c^T$, т.е. чтобы получить f_{A^T} надо перевернуть порядок операций f_A и

транспонировать каждую операцию

$$rb(0) \xleftrightarrow{T} pop()$$

$$a_i * = c \xleftrightarrow{T} a_i \cdot = c$$

$$a_i += a_j * c \xleftrightarrow{T} a_j += a_i \cdot c$$

то есть элементарные операции остаются элементарными, т.е. f_{A^T} работает за c операций

Пример 1: наивное умножение многочленов

$$A(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}; \quad n - \text{параметр}$$

f: $B \mapsto A \cdot B$, где $B = b_0 + \dots + b_{m-1} x^{m-1}$ — линейный оператор от коэф. B с матрицей

транспонированный оператор:

$$\begin{pmatrix} a_0 & 0 & 0 & \dots & 0 \\ a_1 & a_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & \vdots & \vdots & \vdots & a_0 \\ 0 & \vdots & \vdots & \vdots & a_1 \\ & \ddots & \ddots & \ddots & \vdots \\ & & 0 & \vdots & a_{n-1} \end{pmatrix}_{n+m-1} \cdot \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n+m-2} \end{pmatrix}$$
$$= \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} & 0 \\ 0 & a_0 & a_1 & \dots & a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & \vdots & \vdots & a_0 \\ & \ddots & \ddots & \ddots & \vdots \\ & & 0 & \vdots & a_{n-1} \end{pmatrix}_{n+m-1} \cdot \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \\ c_{n+m-2} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n+m-2} \end{pmatrix}$$

Скорость произв. a со всеми подпрограмм c

middle product

Алгоритм:

$\text{mul}_{A,m}(B)$

создаём $n+m-1$ ячеек $c_0, \dots, c_{n+m-2}$

for $i = 0 \dots n-1$

for $j = 0 \dots m-1$

$c_{i+j} += a_i \cdot b_j$

узнаем m ячеек b_j

$\xrightarrow{T}$

$\text{mul}_{A,m}^T(c)$

создаём m ячеек b_j

for $i = n-1 \dots 0$

for $j = m-1 \dots 0$

$b_j += a_i \cdot c_{i+j}$

узнаем $n+m-1$ ячеек $c_0, \dots, c_{n+m-2}$

Пример 2: умножение многочленов

FFT_n - линейный оператор подстановки $\xi^0 \dots \xi^{n-1}$ в $a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$ с матрицей

$$\begin{pmatrix} \xi^{0 \cdot 0} & \xi^{0 \cdot 1} & \dots & \xi^{0 \cdot (n-1)} \\ \xi^{1 \cdot 0} & \xi^{1 \cdot 1} & \dots & \xi^{1 \cdot (n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \xi^{(n-1) \cdot 0} & \xi^{(n-1) \cdot 1} & \dots & \xi^{(n-1) \cdot (n-1)} \end{pmatrix}$$

- симметричная, т.е. $\text{FFT}_n^T = \text{FFT}_n$
аналогично $(\text{FFT}_n^{-1})^T = \text{FFT}_n^{-1}$

Алгоритм:

$\text{mul}_{A,m}(B)$

K - мин число: $2^K \geq n+m-1$

гориз. A 0-м $\text{go } 2^K$

гориз. B 0-м $\text{от } m \text{ go } 2^K$

$\text{fft}(A)$

$\text{fft}(B)$

for $i = 0 \dots 2^K - 1$

$b_i * = a_i$

$\xrightarrow{T}$

$\text{mul}_{A,m}^T(C)$

K - мин число: $2^K \geq n+m-1$

гориз. A 0-м $\text{go } 2^K$

гориз. C 0-м $\text{от } n+m-1 \text{ go } 2^K$

$\text{fft}(A)$

$\text{fft}^{-1}(C)$

for $i = 2^K - 1 \dots 0$

$c_i * = a_i$

$\text{fft}^{-1}(B)$
 одрежуе B от 2^k до $n+m-1$

$\text{fft}(C)$
 одрежуе C от 2^k до m

Работает за 3 вызова fft переке $\min k: 2^k \geq n+m-1 = |C|$

Каково бы работало за 3 fft переке $\min k: 2^k \geq |A|+|C|-1 = 2n+m-2$

Пример 3: multipoint evaluation

$x_0, \dots, x_{m-1}$; n -параметров алгоритма; $P = p_0 + p_1 x + \dots + p_{n-1} x^{n-1}$ — многоч

матрица:

$$\begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^{n-1} \\ x_1^0 & x_1^1 & \dots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m-1}^0 & x_{m-1}^1 & \dots & x_{m-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{pmatrix} = \begin{pmatrix} P(x_0) \\ \vdots \\ P(x_{m-1}) \end{pmatrix}$$

M

$$\begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^{n-1} \\ x_1^0 & x_1^1 & \dots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m-1}^0 & x_{m-1}^1 & \dots & x_{m-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{pmatrix} = \begin{pmatrix} c_0 \cdot x_0^0 + c_1 \cdot x_0^1 + \dots + c_{m-1} \cdot x_0^{m-1} \\ c_0 \cdot x_1^0 + c_1 \cdot x_1^1 + \dots + c_{m-1} \cdot x_1^{m-1} \\ \vdots \\ c_0 \cdot x_{m-1}^0 + c_1 \cdot x_{m-1}^1 + \dots + c_{m-1} \cdot x_{m-1}^{m-1} \end{pmatrix}$$

$$c_i \cdot x_i^j = [t^j] \frac{c_i}{1 - tx_i} \Rightarrow \text{нам надо найти } \sum_{i=0}^{m-1} \frac{c_i}{1 - tx_i} \bmod t^n$$

Алгоритм: в ДД в функции v , соотв $[l, r)$ считаем $\prod_{i=l}^{r-1} (1 - tx_i) = \text{prod}_v$

$$\text{prod}_v = \sum_{i=l}^{r-1} \frac{c_i}{1 - tx_i} = \text{sum}_v \quad \text{prod}_v - \text{константа алгоритма}$$

DDC(v, l, r):

if $r - l = 1$

$$\text{prod}_v = \{1, -x_v\}$$

$$\text{sum}_v = \{c_v\}$$

return

$$m = (l + r) / 2$$

$$\text{DDC}(2v, l, m)$$

$$\text{DDC}(2v+1, m, r)$$

$$\text{prod}_v = \text{prod}_{2v} * \text{prod}_{2v+1}$$

$$\text{sum}_{2v} = \text{mul}_{\text{prod}_{2v+1}, m-l}(\text{sum}_{2v})$$

$$\text{sum}_{2v+1} = \text{mul}_{\text{prod}_{2v}, r-m}(\text{sum}_{2v+1})$$

DDC^T(v, l, r)

if $r - l = 1$

$$\text{ans}_v = \text{sum}_v$$

return

$$m = (l + r) / 2$$

$$\text{sum}_{2v+1} += \text{sum}_v$$

$$\text{sum}_{2v} += \text{sum}_v$$

$$\text{sum}_{2v+1} = \text{mul}_{\text{prod}_{2v}, r-m}^T(\text{sum}_{2v+1})$$

$$\text{sum}_{2v} = \text{mul}_{\text{prod}_{2v+1}, m-l}^T(\text{sum}_{2v})$$

$$\text{DDC}^T(2v+1, m, r)$$

$$\text{DDC}^T(2v, l, m)$$

T
→

$$\text{sum}_v += \text{sum}_{2v}$$

$$\text{sum}_v += \text{sum}_{2v+1}$$

$\text{P\&C}(1, 0, m)$

$$\text{denom} = \text{prod}_1^{-1} \bmod x^n$$

$$\text{sum}_1 = \text{mul}_{\text{denom}, m}(\text{sum}_1)$$

sum_1 упрости с $n+m-1$ до n

$\text{sum}_1 \rightarrow \text{other}$

Начинаю бэ prod_v

$$\text{denom} = \text{prod}_1^{-1} \bmod x^n$$

$\text{sum}_1 \leftarrow \text{вход}$

sum_1 генерирую с n до $n+m-1$

$$\text{sum}_1 = \text{mul}_{\text{denom}, m}^T(\text{sum}_1)$$

$\text{P\&C}^T(1, 0, m)$