Быстрое преобразование Фурье

Вспомним метод Карацубы для умножения двух чисел. Пусть нам даны числа $A = a_0 + xb_0$, $B = a_1 + xb_1$, где $x = 2^{n/2}$, а n – битовая длина записи чисел. Тогда:

$$AB = a_0b_0 + (a_0b_1 + a_1b_0)x + a_1b_1x^2 = a_0b_0 + [(a_0 + a_1)(b_0 + b_1) - a_0b_0 - a_1b_1]x + a_1b_1x^2$$

Так мы сводим умножение двух n-битных чисел к трём умножениям $n/2$-битных чисел. Теперь мы рассмотрим совсем другой подход, позволяющий достигнуть времени $O(n \log n)$. Как мы знаем, любое число $A = a_n a_{n-1} \dots a_1 a_0$ можно представить единственным образом в виде многочлена от 2 с коэффициентами из $\mathbb{Z}_2$:

$$A(2) = a_0 + 2a_1 + 2^2a_2 + \dots + 2^n a_n$$

Тогда естественным способом умножения чисел будет перемножить их многочлены в целых числах, а затем произвести нормировку, т.е. от каждого коэффициента в произведении оставить его остаток по модулю 2, а частное перекинуть в коэффициент при старшей степени.

Вообще, идея перехода к другому представлению одного и того же объекта, в котором какие-то операции будут выполняться проще, имеет очень широкое применение. Так, согласно китайской теореме об остатках, мы можем взять достаточно большое количество модулей p_i и складывать/умножать остатки от деления на эти модули покомпонентно. Также мы сможем проверять числа на равенство. Но некоторые операции при этом станут сложнее – в таком представлении мы не сможем быстро узнавать относительный порядок двух чисел.

Интерполяция многочленов

Широко известный факт: Если у нас есть n пар (x_i, y_i), у которых x_i различны, то мы можем однозначно восстановить многочлен $P(x)$ степени не выше $n-1$ такой что $P(x_i) = y_i$. У этого факта есть два обоснования.

Во-первых, мы можем рассматривать $P(x_i) = y_i$ как n уравнений с n неизвестными, при этом матрицей данного уравнения будет матрица Вандермонда:

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{pmatrix}$$

Её определитель равен $\prod\limits_{j<i}(x_i - x_j)$ и он не нулевой если (и только если) x_i попарно различны.

Это утверждение можно также доказать конструктивно с помощью китайской теоремы об остатках. Кольцо многочленов является евклидовым, то есть, любой многочлен $A(x)$ можно поделить с остатком на многочлен $B(x) \neq 0$:

$$A(x) = D(x) \cdot B(x) + R(x),\ \deg R(x) < \deg B(x)$$

Здесь $\deg P$ – степень многочлена, т.е. номер наибольшей степени, при которой стоит ненулевой коэффициент. Считаем, что $\deg 0 = -\infty$. Записанный здесь $R(x)$ называют остатком от деления многочлена A на B. В этих терминах можно заметить, что $P(x) \equiv P(a) \pmod{x - a}$.

1. (1б) Обоснуйте это.

Исходя из этого можно явно записать интерполирующий многочлен (многочлен Лагранжа). Считаем, что нам дана система $P(x_i) = y_i \pmod{x - x_i}$. Так как $x - x_i$ и $x - x_j$ взаимно просты если x_i и

x_j различны, по китайской теореме мы можем восстановить многочлен по модулю $\prod_{i=1}^{n}(x-x_i)$, при этом степень полученного многочлена будет не больше $n-1$, что значит, что он будет решением задачи. Получим:

$$P(x) = \sum_{i=1}^{n} y_i M_i (M_i^{-1} \bmod x - x_i) = \sum_{i=1}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

Таким образом, мы можем переключаться между представлением многочлена как последовательности коэффициентов и его представлением в виде набора значений в различных точках. При этом во втором представлении мы можем умножать их за $O(n)$ арифметических операций.

Кажется, мы ничего не выиграли, т.к. как перевод многочлена в это представление, так и возврат из него, достаточно трудоёмки. Но у нас появилось пространство для манёвра – мы можем сами выбрать набор x_i.

Комплексные корни из единицы

Мы будем говорить об алгоритмах, работающих с вещественными числами. Чтобы иметь возможность сосредоточиться на сути алгоритмов, а не технических деталях, мы будем при анализе асимптотики учитывать только число проделанных арифметических операций.

Позже мы увидим, что интерполяция многочлена может быть произведена эффективно если x_i образуют циклическую группу по умножению. Если группа имеет размер n, то для всех таких элементов должно быть выполнено $x_i^n = x_i^0 = 1$. В вещественных числах такое уравнение в принципе может иметь только два решения, это 1 и -1 если n чётное число.

У нас есть два наиболее простых способа решить эту проблему – искать такие числа в кольцах остатков или в комплексных числах. В комплексных числах имеет место формула Эйлера:

$$e^{i\varphi} = \cos\varphi + i \sin\varphi$$

Решая с её помощью уравнение $x^n = 1$ и считая, что $x = e^{a+ib}$ приходим к системе:

$$\begin{cases} e^{an} = 1, \\ \cos bn = 1, \\ \sin bn = 0 \end{cases} \implies \begin{cases} a = 0, \\ b = \dfrac{2\pi k}{n}, \quad k = 0, 1, \dots, n-1 \end{cases}$$

Это является частным случаем Формулы Муавра

Пусть теперь ω_n – образующий элемент в нашей мультипликативной группе, то есть,

$$\omega_n^k = 1 \iff k \equiv 0 \pmod n$$

При этом будем считать, что $n = 2^k$. Если мы хотим только умножать многочлены, это нам подойдёт, т.к. к такому случаю можно прийти просто добавив нулевых коэффициентов в представление. Пусть $P(x) = \sum\limits_{i=0}^{2^k-1} a_i x^i$. Введём в рассмотрение многочлены, составленные из чётных и нечётных коэффициентов соответственно:

$$A(x) = \sum_{i=0}^{2^{k-1}-1} a_{2i} x^i, \quad B(x) = \sum_{i=0}^{2^{k-1}-1} a_{2i+1} x^i$$

Тогда можно записать $P(x) = A(x^2) + xB(x^2)$. Воспользуемся следующим фактом: $\omega_n^2 = \omega_{n/2}$.

2. (1б) Докажите это.

Это значит, что мы можем свести вычисление $P(w_n^k)$ к вычислениям $A(w_{n/2}^k)$ и $B(w_{n/2}^k)$, то есть, к двум задачам, размер которых в два раза меньше. Итого получим следующую формулу для пересчёта:

$$P(w_n^k) = A\left(\omega_{n/2}^k\right) + \omega_n^k \cdot B\left(\omega_{n/2}^k\right), \; k = 0, 1, \ldots, n-1$$

Время выполнения полученной процедуры оценивается как:

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) = O(n \log n)$$

Наконец, отметим, что поиск значений многочлена в комплексных корнях из единицы, то есть, переход от последовательности $\{a_j\}_{j=0}^{n-1}$ к последовательности $\{\widetilde{a_k}\}_{k=0}^{n-1}$, определённой как:

$$\widetilde{a_k} = \sum_{j=0}^{n-1} a_j \omega_n^{jk}$$

Называется *дискретным преобразованием Фурье*. А если речь заходит о методах быстрого его вычисления, то тут уже говорят *быстрое преобразование Фурье*. В частности, выше описан алгоритм Кули-Тьюки.

3. (3б) Вычислите преобразование Фурье по методу Кули-Тьюки от:

1. Последовательности $\{a_i\}_{i=0}^7 = \{1, 2, 3, 4, 1, 2, 3, 4\}$
2. Последовательности $\{b_i\}_{i=0}^7 = \{1, 1, 1, 1, 1, 1, 1, 1\}$

Указание: $\omega_8 = \exp\left(\frac{\pi i}{4}\right) = \cos\frac{\pi}{4} + i \sin\frac{\pi}{4} = \frac{(1+i)\sqrt{2}}{2}$

Обратное преобразование

После того, как мы научились вычислять значения многочлена в нескольких точках, нужно научиться переходить обратно, т.е. интерполировать многочлен. Это можно делать сразу, если заметить, что все операции, которые мы делали, были обратимыми – мы переупорядочивали элементы последовательности, а также применяли преобразование $P(w_n^k) = A\left(\omega_{n/2}^k\right) + \omega_n^k \cdot B\left(\omega_{n/2}^k\right)$. Если обозначить $t = n/2$, получим обратное выражение:

$$\begin{cases} P(w_n^k) &= A\left(\omega_t^k\right) + \omega_n^k \cdot B\left(\omega_t^k\right) \\ P(\omega_n^{k+t}) &= A\left(\omega_t^k\right) - \omega_n^k \cdot B\left(\omega_t^k\right) \end{cases} \Longrightarrow \begin{cases} A(\omega_t^k) &= \dfrac{P(\omega_n^k) + P(\omega_n^{k+t})}{2} \\ B(\omega_t^k) &= \dfrac{P(\omega_n^k) - P(\omega_n^{k+t})}{2\omega_n^k} \end{cases}$$

С другой стороны, можно работать непосредственно с матрицей преобразования:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Здесь $x_i = w_n^{i-1}$, то есть, $a_{ij} = \omega_n^{ij}$ (в 0-индексации). Найдём обратную к ней, т.е., матрицу b_{kj} такую, что:

$$\sum_{k=0}^{n-1} \omega_n^{ik} b_{kj} = \delta_{ij}$$

Положим $b_{kj} = \omega_n^{-kj}$. Тогда:

$$\sum_{k=0}^{n-1} \omega_n^{ik} \omega_n^{-kj} = \sum_{k=0}^{n-1} \omega_n^{(i-j)k} = n \cdot \delta_{ij}$$

4. (1б) Обоснуйте последнее равенство.

Отсюда следует, что обратная матрица задаётся как $b_{kj} = \frac{1}{n} w_n^{-kj}$. Значит, для обратного преобразования нам нужно проделать ту же процедуру, но вместо w_n и его степеней использовать степени сопряжённого к нему элемент w_n^{-1}, а в конце разделить результат на n.

5. (2б) Вычислите обратное преобразование Фурье массива:

$$A = \{10,\ 2 + (3\sqrt{2} + 2)i,\ 0,\ 2 + (3\sqrt{2} - 2)i,\ -2,\ 2 + (2 - 3\sqrt{2})i,\ 0,\ 2 - (2 + 3\sqrt{2})i\}$$

6. (3б) Найдите произведение многочленов $A(x) = 3x + 2$ и $B(x) = x^2 + 1$, используя FFT.

7. (3б) Дано множество различных чисел $A \subset \{1, \ldots, m\}$. Рассмотрим множество $A + A$, образованное попарными суммами элементов A. Докажите, что существует процедура построения множества $A + A$ за $o(m^2)$.

Нерекурсивный алгоритм

Для начала приведём общий алгоритм в виде псевдокода.

```
function fft(array a):
    n = length(a)
    if n = 1:
        return a
    A = fft({a_0, a_2, ..., a_{n-2}})
    B = fft({a_1, a_3, ..., a_{n-1}})
    P = array(n)
    for i from 0 to n - 1:
        P[i] = A[i mod (n / 2)] + ω_n^i * B[i mod (n / 2)]
    return a
```

Здесь n – степень двойки, ω_n – образующий корень из единицы степени n. Разберём схему Кули-Тьюки подробно на примере последовательности $a_0, a_1, \ldots, a_7$. Вначале мы строим дерево, разбивая последовательность на чётные и нечётные элементы:

<table>
<tr><td colspan="8">$a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$</td></tr>
<tr><td colspan="4">$a_0 + a_2x + a_4x^2 + a_6x^3$</td><td colspan="4">$a_1 + a_3x + a_5x^2 + a_7x^3$</td></tr>
<tr><td colspan="2">$a_0 + a_4x$</td><td colspan="2">$a_2 + a_6x$</td><td colspan="2">$a_1 + a_5x$</td><td colspan="2">$a_3 + a_7x$</td></tr>
<tr><td>a_0</td><td>a_4</td><td>a_2</td><td>a_6</td><td>a_1</td><td>a_5</td><td>a_3</td><td>a_7</td></tr>
</table>

Обратим внимание на двоичные записи индексов, которые мы получили на нижнем уровне:

$$\begin{array}{lcl} 0 = 000_2 & , & 1 = 001_2 \\ 4 = 100_2 & , & 5 = 101_2 \\ 2 = 010_2 & , & 3 = 011_2 \\ 6 = 110_2 & , & 7 = 111_2 \end{array}$$

Это соответствует последовательности чисел $\{0, 1, 2, \ldots, 7\}$ чья двоичная запись развернута. Такое наблюдение позволяет находить последовательность на нижнем уровне непосредственно, не проводя все рекурсивные действия. Это служит базой нерекурсивного алгоритма.

8. (2б) Обобщите это наблюдение для случая $n = 2^k$ и докажите его.

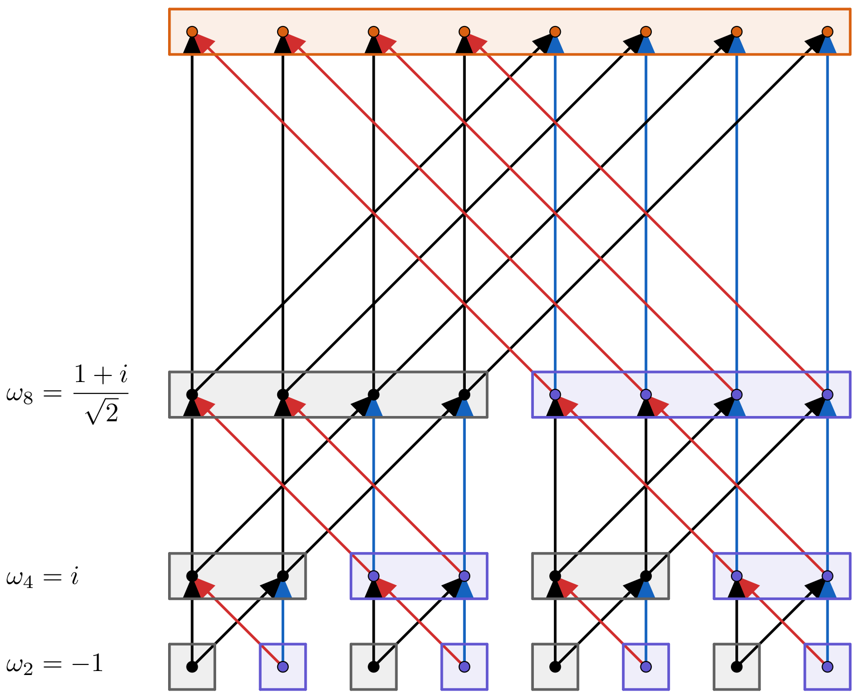
Теперь мы должны, начиная с нижнего уровня таблицы, заменять содержимое ячеек на преобразование Фурье соответствующих многочленов. Нижний уровень у нас уже есть, т.к. в нём находятся многочлены нулевой степени.

$$P(\omega_n^k) = A(\omega_{n/2}^k) + \omega_n^k B(\omega_{n/2}^k)$$

При ручном вычислении, учитывая, что $\omega_n^{n/2} = -1$, её проще воспринимать в следующем виде:

$$\begin{cases} P(\omega_n^k) = A(\omega_{n/2}^k) + \omega_n^k B(\omega_{n/2}^k), \\ P(\omega_n^{k+n/2}) = A(\omega_{n/2}^k) - \omega_n^k B(\omega_{n/2}^k) \end{cases}, 0 \le k < n/2$$

То есть, элементы, соответствующие одному и тому же $\omega_{n/2}^k$ в A и B будут учитываться в $P(\omega_n^k)$ и $P(\omega_n^{k+n/2})$. При этом $A(\omega_{n/2}^k)$ в обоих случаях будет иметь коэффициент 1, а $B(\omega_{n/2}^k)$ будет иметь коэффициент ω_n^k в первом случае и $-\omega_n^k$ во втором. Это удобно изображать графически в виде диаграммы, приведённой ниже. Элементарный шаг при переходе на следующий уровень обычно называют схемой бабочки из-за визуальной схожести.



Здесь чёрные блоки точек – соответствующие чётным коэффициентам A, фиолетовые блоки – соответствующие нечётным коэффициентам B. Соответственно, стрелками указаны переходы в $P(\omega_n^k)$ и $P(\omega_n^{k+n/2})$. При этом чёрный цвет стрелки означает коэффициент 1, красный цвет – коэффициент ω_n^k, а синий – коэффициент $-\omega_n^k$. Приведём общий алгоритм для умножения двух многочленов с помощью преобразования Фурье.

1. Даны многочлены A и B, $\deg A = n_1$ и $\deg B = n_2$. Найдём k такой что $n = 2^k \geq n_1 + n_2 + 1$.
2. Посчитаем преобразование Фурье A и B, используя ω_n в качестве образующего.
3. Покомпонентно перемножим получившиеся последовательности, обозначим результат P.
4. Посчитаем преобразование Фурье от P, используя ω_n^{-1} и разделим все коэффициенты на n.

9. (1б) Постройте схему бабочки для перехода от ω_8 к ω_{16}. Нижние уровни можно не приводить.

10. (1б) Посчитайте преобразование Фурье для многочлена $1 + x + x^5 + x^6$, считая $n = 8$.

11. (2б) Посчитайте произведение $1 + 2x$ и $3 + 4x^3$ с помощью быстрого преобразования Фурье.

12. (2б) Пусть $P(x) = a_0 + a_1 x + \ldots a_{n-1} x^{n-1}$. Введём оператор DFT:

$$\mathrm{DFT}_n(A) : \{a_0, a_1, \ldots, a_{n-1}\} \to \{P(\omega_n^0), P(\omega_n^1), \ldots, P(\omega_n^{n-1})\}$$

Также введём оператор разворота:

$$\mathrm{rev}(A) : \{a_0, a_1, a_2 \ldots, a_{n-1}\} \to \{a_0, a_{n-1}, \ldots, a_2, a_1\}$$

Покажите, что для двойного преобразования Фурье имеет место равенство:

$$\mathrm{DFT}_n \cdot \mathrm{DFT}_n(A) = n \cdot \mathrm{rev}(A)$$

Из этого будет следовать, что $\mathrm{DFT}_n^{-1}(A) = n^{-1} \cdot \mathrm{rev} \cdot \mathrm{DFT}_n(A)$*, т.к.* $\mathrm{rev}^{-1}(A) = \mathrm{rev}(A)$*. То есть, при обратном преобразовании вместо* ω_n^{-1} *можно использовать тот же* ω_n*, а затем развернуть отрезок последовательности с* 1 *по* $n-1$ *элемент и разделить всё на* n*.*

13. (1б) DFT_n может быть задан некоторой матрицей, действующей на векторы. Выпишите матрицы, соответствующие операторам DFT_n и DFT_n^{-1}.

В следующих двух задачах мы будем применять преобразование к вектор-функциям. Следует считать, что оно задаётся умножением вектор-функции на соответствующую матрицу.

1*. (2б) Пусть $f(x) = \sum\limits_{k=0}^{\infty} a_k x^k$. Введём оператор "среза":

$$\mathrm{slice}_{n,r}(f) : \sum_{i=0}^{\infty} a_i x^i \to \sum_{k=0}^{\infty} a_{n \cdot k + r}\ x^{n \cdot k + r}$$

Покажите, что:

$$\begin{pmatrix} \mathrm{slice}_{n,0}(f) \\ \mathrm{slice}_{n,1}(f) \\ \vdots \\ \mathrm{slice}_{n,n-1}(f) \end{pmatrix} = \mathrm{DFT}_n^{-1} \begin{pmatrix} f(\omega_n^0 x) \\ f(\omega_n^1 x) \\ \vdots \\ f(\omega_n^{n-1} x) \end{pmatrix}$$

Смысл данного равенства в том, что оно позволяет нам в явном виде выделять производящие функции для подпоследовательностей, образующих арифметическую прогрессию с шагом n*. В частности, при* $n = 2$ *мы сможем разделить последовательность на чётные и нечётные элементы, и нам даже не понадобится использовать комплексные числа для этого.*

14. (1б) Рассмотрим обозначения прошлой задачи. Посчитайте $\mathrm{DFT}_2^{-1}\{f(x),\ f(-x)\}$.
Покажите, что полученные выражения соответствуют $\mathrm{slice}_{2,0}(f)$ и $\mathrm{slice}_{2,1}(f)$.