Базовое FFT

ЛКШ 2026

19.07.2026

1. Пререквизиты

Перед самим рассказом про алгоритм быстрого преобразования Фурье придется вспомнить про несколько (относительно) простых вещей.

1.1. Комплексные числа

Всем вам известно, что такое вещественные числа. Обычно их поле обозначают буквой $\mathbb{R}$. Однако у них есть (не)большой минус — не у каждого многочлена есть вещественный корень (если говорить по-умному, то $\mathbb{R}$ не является алгебраически замкнутым полем) — например, у $x^2 + 1$.

Однако люди изобрели решение этой проблемы — можно придумать волшебную букву i, для которой верно $i^2 + 1 = 0$. Так появляются числа вида $a + bi$ — их называют комплексными. Такое поле обозначают $\mathbb{C}$. По сути, комплексные числа — это просто пары из двух вещественных. Даже понятно, как делать с ними арифметические операции:

- $(a + bi) \pm (c + di) = (a \pm c) + (b \pm d)i$
- $(a + bi) \cdot (c + di) = (ac - bd) + (ad + bc)i$
- $(a + bi)^{-1} = \frac{a-bi}{a^2+b^2}$
  (конечно же, $a + bi \neq 0$)

Пара нетривиальных операций:

- Модуль
  $|a + bi| = \sqrt{a^2 + b^2}$
- Сопряжение
  $\overline{a + bi} = a - bi$
  Funfact: $z \cdot \overline{z} = |z|^2$

Существует основная теорема алгебры, которая говорит нам, что у любого неконстантного многочлена с комплексными коэффициентами есть комплексный корень (более подробно вам расскажут о ней в университете, мы же в нее просто поверим). Благодаря этому получаем полезное следствие — любой многочлен раскладывается на линейные множители, а значит у многочлена степени n есть ровно n корней (с учетом кратности).

На самом деле, гораздо проще считать комплексное число $(a+bi)$ точкой (a, b) на обычной плоскости. Вместо Ox у нас Re — вещественная часть, а вместо Oy у нас Im — мнимая часть. Чтобы лучше понять, как работает умножение, представим комплексное число в *тригонометрической форме* (для удобства $z \neq 0$):

$$z = |z|(\cos\varphi + i\sin\varphi)$$

$|z|$ называют *модулем*, а φ — *аргументом*. Заметим, что для φ и $\varphi + 2\pi$ получаются одинаковые комплексные числа, так что во избежание путаницы есть *главный аргумент* — φ в $[0, 2\pi)$.

Тогда умножение предстает в ином виде:

$$z_1 \cdot z_2 = |z_1|(\cos\alpha + i\sin\alpha) \cdot |z_2|(\cos\beta + i\sin\beta) =$$
$$= |z_1||z_2|((\cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta) + i(\cos\alpha \cdot \sin\beta + \sin\alpha \cdot \cos\beta)) =$$
$$= |z_1||z_2|(\cos(\alpha+\beta) + i\sin(\alpha+\beta))$$

Таким образом, при умножении модули чисел перемножаются, а аргументы складываются. Даже сошлось с обычным умножением вещественных чисел)

1.2. Первообразный корень из единицы

В случае вещественных чисел уравнение $x^n = 1$ имеет одно или два решения для натуральных n. Но как мы знаем, в комплексных числах должны быть все n решений. Попытаемся найти их вид:

- Мы знаем, что модули чисел при умножении перемножаются, а значит, чтобы узнать модуль корня, надо решить $x^n = 1$, где $x \geq 0$ и $x \in \mathbb{R}$. Значит $x = 1$ и все корни такого уравнения лежат на окружности радиусом 1.

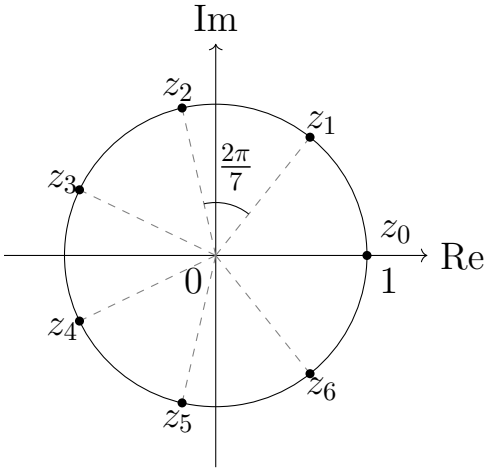
- Пусть аргумент корня — φ.

  У числа 1 главный аргумент 0, поэтому $n \cdot \varphi = 0 + 2\pi k$, где $k \in \mathbb{Z}$

  Значит $\varphi = \dfrac{2\pi k}{n}$

  Нетрудно убедиться, что при $k \in \{0, \ldots, n-1\}$ числа для каждого φ разные, а уже с $k = n$ числа повторяются (действительно, изменение аргумента на $2\pi k$ не меняет число).

Поэтому все корни $x^n = 1$ имеют вид $z_k = \cos\frac{2\pi k}{n} + i\sin\frac{2\pi k}{n}$, $k \in \{0, 1, \ldots, n-1\}$.
Нарисуем их расположение для $n = 7$:



Нетрудно убедиться, что если ξ_1 и ξ_2 — корни x^n-1, то $\xi_1 \cdot \xi_2$ — тоже. А значит множество корней замкнуто относительно умножения.

Обозначим $\xi_1 = \cos\dfrac{2\pi}{n} + i\sin\dfrac{2\pi}{n}$ (z_1 на рисунке для $n=7$). Тогда все остальные корни можно получить в виде $\xi_k = (\xi_1)^k$, где $k \in \{0, 1, \ldots, n-1\}$. То есть один корень *порождает* все остальные корни.

Давайте назовем его одним из первообразных корней из единицы. Вообще, первообразный корень из единицы — это такой корень n-ой степени из единицы ξ, что $\xi^k \neq 1$ для всевозможных $k \in \{1, 2, \ldots, n-1\}$. Мы уже выяснили, что ξ_1 нам подойдет. На самом деле, подойдут все ξ_k такие, что $(k, n) = 1$ и только они (данная задача оставляется читателю в качестве упражнения).

На самом деле, все это можно было пропустить словами: “Корни из единицы — это циклическая группа”. Если не знаете, что такое группа, то можете погуглить, если хотите, но в дальнейшем нам это не понадобится (скорее всего, вы узнаете эти непонятные слова получше на курсе алгебры в университете).

1.3. Деление многочлена на многочлен

Быстро напомним, что при делении многочлена $P(x)$ на $g(x) \neq 0$ определены частное $Q(x)$ и остаток $R(x)$, причем $\deg R(x) < \deg g(x)$ и верно:

$$P(x) = g(x) \cdot Q(x) + R(x)$$

Нетрудно убедиться, что $Q(x)$ и $R(x)$ определяются единственным образом. Действительно, пусть

$$\begin{gathered} g(x) \cdot Q_1(x) + R_1(x) = P(x) = g(x) \cdot Q_2(x) + R_2(x) \\ \Downarrow \\ g(x)(Q_1(x) - Q_2(x)) = R_2(x) - R_1(x) \end{gathered}$$

Если $Q_1(x) = Q_2(x)$, то и $R_1(x) = R_2(x)$. Иначе слева написан ненулевой многочлен степени хотя бы $\deg g(x)$, а справа — либо нулевой многочлен, либо многочлен степени менее $\deg g(x)$ — противоречие.

Существование тривиально следует из деления в столбик.

1.4. Обратный многочлен по модулю

Хотим найти $B(x)$ такой, что $A(x)B(x) \equiv 1 \pmod{C(x)}$ при условии того, что $A(x)$ и $C(x)$ взаимно просты.

Решение никак не отличается от целых чисел, ведь расширенный алгоритм Евклида работает здесь так же. Решим $u(x)A(x) + v(x)C(x) = (A(x), C(x)) = 1$ (на самом деле НОД многочленов определен с точностью до константы, но это просто исправимо), положим $B(x) = u(x)$.

1.5. Теорема Безу

В будущем нам потребуется узнавать значение многочлена в некоторой точке, а с этим прекрасно помогает теорема Безу:

Теорема 1.1: Теорема Безу

Остаток от деления многочлена $P(x)$ на $(x-a)$ равен $P(a)$.

Пусть $Q(x)$ — это частное от деления $P(x)$ на $(x-a)$, а $R(x)$ — остаток:

$$P(x) = Q(x) \cdot (x-a) + R(x)$$

Поскольку $\deg R(x) < \deg (x-a) = 1$, $R(x)$ является константой, поэтому

$$P(x) = Q(x) \cdot (x-a) + R$$

Подставим $x = a$, получим искомое равенство:

$$P(a) = Q(a) \cdot 0 + R = R$$

1.6. КТО для многочленов

Теорема 1.2: Китайская теорема об остатках для многочленов

Пусть $P_1(x), P_2(x), \ldots, P_n(x)$ — попарно взаимно простые многочлены. Тогда для любых $r_1(x), \ldots, r_n(x)$ существует единственный многочлен $F(x)$, удовлетворяющий системе:

$$\begin{cases} F(x) \equiv r_1(x) \pmod{P_1(x)} \\ F(x) \equiv r_2(x) \pmod{P_2(x)} \\ \ldots \\ F(x) \equiv r_n(x) \pmod{P_n(x)} \\ \deg F(x) < \deg P_1(x) + \deg P_2(x) + \ldots + \deg P_n(x) \end{cases}$$

Обозначим $M(x) = P_1(x) \cdot P_2(x) \cdot \ldots \cdot P_n(x)$.

- Докажем единственность решения:

  Пусть $F_1(x)$ и $F_2(x)$ — решения данной системы.

  Тогда $F_1(x) - F_2(x)$ кратен $M(x)$ (в силу взаимной простоты P_i).

  Но $\deg (F_1(x) - F_2(x)) < \deg M(x)$

  А значит это нулевой многочлен, то есть $F_1(x) = F_2(x)$

- Покажем существование:

  Пусть $A_i(x) = \frac{M(x)}{P_i(x)}$, (заметим, что здесь имеется в виду деление нацело, так как $M(x)$ делится на $P_i(x)$) а $B_i(x)$ — многочлен обратный к A_i по модулю P_i (именно здесь нам нужна взаимная простота P_i). Тогда

$$A_j(x) \cdot B_j(x) \equiv \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \pmod{P_i(x)}$$

  $F = \sum_{i=1}^{n} r_i(x) \cdot A_i(x) \cdot B_i(x)$ подойдет, остается взять его по модулю $M(x)$.

2. FFT

Кода здесь намеренно не будет, лучше задавайте вопросы)

2.1. Что хотим делать?

Заметим, что из КТО и теоремы Безу сразу следует тот факт, что многочлен степени не более n однозначно восстанавливается по своим значениям в $n + 1$ различной точке. Этот факт потребуется нам для умножения многочленов:

- Берем многочлен $a(x)$ степени n

  Берем многочлен $b(x)$ степени m

- Берем $n + m + 1$ различную точку: $x_1, x_2, \ldots, x_{n+m+1}$

- Считаем $A_i = a(x_i)$

  Считаем $B_i = b(x_i)$

- Считаем $C_i = A_i \cdot B_i$

- Как-то восстанавливаем $c(x) = a(x) \cdot b(x)$ по C_i

  $\deg c = \deg a + \deg b = n + m$, так что нам хватит $n + m + 1$ точки.

В данном плане непонятно как делать две вещи:

1. Достаточно быстро считать A_i и B_i
2. Достаточно быстро восстанавливать $c(x)$ по C_i

Оказывается, что если выбрать хорошие x_i, то два вышеописанных действия можно сделать довольно быстро.

В качестве x_i мы возьмем корни из единицы степени 2^t, где $2^t \geq n + m + 1$.

Теперь нужно понять, как достаточно быстро посчитать значения многочлена в корнях из единицы. Именно этим и занимается FFT (точнее, само преобразование "коэффициенты $\to$ значения" называется DFT — Discrete Fourier Transform, а FFT — это быстрый алгоритм его вычисления).

2.2. Вариант 1

Для нахождения $P(\xi_i)$ мы будем искать $P(x) \bmod (x - \xi_i)$.

Зафиксируем степень многочлена: $\deg P(x) = n - 1 = 2^t - 1$ (то есть у многочлена 2^t коэффициентов). Если степень $P(x)$ не является такой, то будем считать, что несколько старших членов — нулевые. Это увеличит число коэффициентов не более чем вдвое, так что асимптотика не ухудшится.

Заметим, что мы можем взять $P(x)$ по модулю $(x - a)$ за $O(n)$, однако если сделать так для всех корней, будет слишком долго.

Давайте будем делать так:

- Хотим посчитать $P(x)$ по модулям $(x-a)$, $(x-b)$, $(x-c)$, $(x-d)$, и a, b, c, d — различные.
- $P(x) \to P(x) \bmod (x-a)(x-b)(x-c)(x-d)$
- 1. $P(x) \bmod (x-a) = (P(x) \bmod (x-a)(x-b)) \bmod (x-a)$
  2. $P(x) \bmod (x-b) = (P(x) \bmod (x-a)(x-b)) \bmod (x-b)$
  3. $P(x) \bmod (x-c) = (P(x) \bmod (x-c)(x-d)) \bmod (x-c)$
  4. $P(x) \bmod (x-d) = (P(x) \bmod (x-c)(x-d)) \bmod (x-d)$

Плюсы: обобщается на любую степень двойки

Минусы: не совсем понятно, как искать остаток многочлена по какому-то другому непонятному многочлену за быстро.

Мы воспользуемся тем, что у нас не случайные a, b, c, d, а именно корни из единицы.

Заметим, что взять $P(x)$ по модулю $(x^k - a)$ можно за $O(n)$ — простым делением в столбик.

Для начала вспомним, что ξ_1 — первообразный корень и верно

$$(x - \xi_1) \cdot (x - \xi_2) \cdot \ldots \cdot (x - \xi_n) = x^n - 1$$

Поскольку n — степень двойки, то

$$x^{2^t} - 1 = \left(x^{2^{t-1}} - 1\right)\left(x^{2^{t-1}} + 1\right)$$

$x^{2^{t-1}} - 1$ по сути такая же скобка, что и $x^{2^t} - 1$, так что можно было бы свести задачу рекурсивно. А с $x^{2^{t-1}} + 1$ не совсем удобно — делать разность квадратов с $x^2 + a^2$ довольно муторно. Посмотрим с другой стороны: $x^{2^{t-1}} + 1 = x^{2^{t-1}} - (-1)$, причем -1 тоже является корнем из единицы.

Это наталкивает на мысль: в каждый момент времени в нашей разделяйке будем поддерживать $P(x)$ по модулю $x^m - \xi_1^k$, где m — степень двойки, а k — четное число. Тогда, чтобы спуститься ниже в рекурсию, нам потребуется понять, что:

$$x^m - \xi_1^k = \left(x^{\frac{m}{2}} - \xi_1^{\frac{k}{2}}\right)\left(x^{\frac{m}{2}} + \xi_1^{\frac{k}{2}}\right) = \left(x^{\frac{m}{2}} - \xi_1^{\frac{k}{2}}\right)\left(x^{\frac{m}{2}} - (-1)\cdot\xi_1^{\frac{k}{2}}\right) =$$
$$= \left(x^{\frac{m}{2}} - \xi_1^{\frac{k}{2}}\right)\left(x^{\frac{m}{2}} - \xi_1^{\frac{n}{2}}\cdot\xi_1^{\frac{k}{2}}\right) = \left(x^{\frac{m}{2}} - \xi_1^{\frac{k}{2}}\right)\left(x^{\frac{m}{2}} - \xi_1^{\frac{k+n}{2}}\right)$$

Остается подобрать стартовый показатель k, с которого мы начнем (а именно, $(x^n - \xi_1^k)$), так, чтобы при каждом спуске новые показатели $\frac{k}{2}$ и $\frac{k+n}{2}$ оставались четными (на самом нижнем уровне это уже не важно). Посмотрев на битовую запись, нетрудно убедиться, что $k = 0$ подойдет.

Таким образом, если в текущий момент разделяйки мы храним в рекурсии $P(x) \bmod (x^m - \xi_1^k)$, то в ветках рекурсии мы будем хранить по модулям:

1. $x^{\frac{m}{2}} - \xi_1^{\frac{k}{2}}$
2. $x^{\frac{m}{2}} - \xi_1^{\frac{k+n}{2}}$

При этом переход к детям довольно простой. Пусть $root = \xi_1^{\frac{k}{2}}$. Степень текущего рассматриваемого многочлена меньше m (у него m коэффициентов). Пусть $a_0, \ldots, a_{\frac{m}{2}-1}$ — младшая половина его коэффициентов, а $b_0, \ldots, b_{\frac{m}{2}-1}$ — старшая половина. Тогда в первую половину мы передадим многочлен с коэффициентами: $u_i = a_i + b_i \cdot root$, а во вторую — с $v_i = a_i - b_i \cdot root$ (следует напрямую из деления в столбик). Заметим, что это можно делать ***inplace***.

Стоит помнить, что по итогу мы получим значения в корнях из единицы не по порядку, а в порядке бит-реверса.

Для того, чтобы получить обратное FFT, достаточно просто обратить все наши действия: зная u_i и v_i, восстанавливаем $a_i = \dfrac{u_i + v_i}{2}$ и $b_i = \dfrac{u_i - v_i}{2 \cdot root}$, поднимаясь от листьев рекурсии к корню.

Получаем алгоритм с асимптотикой $O(n \log n)$. К сожалению, тут надо дольше объяснять, чем во втором варианте, однако мне кажется, что так понять FFT в итоге гораздо проще.

Так мы получаем реализацию FFT в 15–20 строчек, и скопированное обратное FFT (IFFT) такой же длины)

2.3. Вариант 2

Здесь мы обратимся к магии формул.

Снова будем считать, что $\deg P(x) = n - 1 = 2^t - 1$.

Заметим, что:

$$P(x) = \sum_{k=0}^{n-1} a_k x^k = \sum_{k=0}^{n/2-1} a_{2k} x^{2k} + x \sum_{k=0}^{n/2-1} a_{2k+1} x^{2k}$$

Пусть $A(x) = \sum\limits_{k=0}^{n/2-1} a_{2k} x^k$, $B(x) = \sum\limits_{k=0}^{n/2-1} a_{2k+1} x^k$.

Тогда $P(x) = A(x^2) + xB(x^2)$.

То есть $P(\xi) = A(\xi^2) + \xi B(\xi^2)$.

Заметим, что если ξ — первообразный корень степени n, то ξ^2 — это первообразный корень степени $\frac{n}{2}$. Давайте рекурсивно посчитаем FFT от $A(x)$ и $B(x)$ — получим значения $A_k = A(\xi^{2k})$ и $B_k = B(\xi^{2k})$, после чего $(0 \le k < \frac{n}{2})$:

- $P_k = A_k + \xi^k \cdot B_k$
- $P_{k+\frac{n}{2}} = A(\xi^{2(k+\frac{n}{2})}) + \xi^{k+\frac{n}{2}} \cdot B(\xi^{2(k+\frac{n}{2})}) = A(\xi^{2k}) - \xi^k \cdot B(\xi^{2k}) = A_k - \xi^k \cdot B_k$

Очевидно, что такой алгоритм тоже работает за $O(n \log n)$.

Пока что можем реализовать IFFT через обращение действий FFT.

Такая рекурсивная реализация довольно долгая — каждый раз мы передаем вниз два новых вектора. Давайте заранее расположим коэффициенты так, чтобы на каждом этапе рекурсии нам приходилось бы обрабатывать непрерывный отрезок коэффициентов.

Сначала в левую ветку рекурсии отправляются все коэффициенты с четными индексами, а в правую — с нечетными. Потом и там, и там в левую ветку пойдут все элементы, у которых предпоследний бит нулевой, а в правую — у которых единичный. И так далее. В конце концов все коэффициенты расположатся в порядке бит-реверса. Поэтому давайте заранее расположим коэффициенты в таком порядке — после этого нам достаточно пройтись по всем уровням рекурсии обычными for-ами.

Пусть ξ — первообразный корень и $F_k = P(\xi^k) = \sum\limits_{i=0}^{n-1} a_i (\xi^k)^i = \sum\limits_{i=0}^{n-1} a_i \xi^{ki}$

Рассмотрим многочлен $F = \sum\limits_{i=0}^{n-1} F_i x^i$

Теперь почему-то попытаемся подставить туда ξ^{-k}:

$$F(\xi^{-k}) = \sum_{i=0}^{n-1} F_i (\xi^{-k})^i = \sum_{i=0}^{n-1} \left(\sum_{j=0}^{n-1} a_j \xi^{ij} \right) \xi^{-ki} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j \xi^{i(j-k)} =$$

$$= \sum_{j=0}^{n-1} a_j \sum_{i=0}^{n-1} \xi^{i(j-k)}$$

Если $j = k$, то вторая сумма схлопнется в n и получится слагаемое na_k.

Иначе вторая сумма является геометрической прогрессией и равна:

$$\frac{\xi^{n(j-k)} - 1}{\xi^{j-k} - 1} = \frac{1^{j-k} - 1}{\xi^{j-k} - 1} = 0$$

Знаменатель не 0, так как $j \neq k$, а ξ — первообразный корень.

Таким образом $F(\xi^{-k}) = na_k$

То есть чтобы посчитать IFFT, нужно запустить FFT, взяв за первообразный корень обратный к тому, с помощью которого мы считали FFT, а потом в конце поделив на n.

На самом деле, можно никак не думать про ξ и ξ^{-1}, если воспользоваться еще одним забавным фактом:

$$P(\xi^{-k}) = \sum_{i=0}^{n-1} a_i(\xi^{-k})^i = a_0 + \sum_{i=1}^{n-1} a_i\xi^{-ki} = a_0 + \sum_{i=1}^{n-1} a_i\xi^{kn-ki} =$$
$$= a_0 + \sum_{i=1}^{n-1} a_i\xi^{k(n-i)} = a_0 + \sum_{i=1}^{n-1} a_{n-i}\xi^{ki} = a_0 + \sum_{i=1}^{n-1} a_{n-i}(\xi^k)^i$$

То есть вместо запуска FFT с первообразным корнем ξ^{-1} можно просто развернуть коэффициенты с индексами $1, 2, \ldots, n-1$ и вызвать обычное FFT.

Дальнейшие оптимизации можете прочитать в прикрепленном файле.

3. NTT

А что если мы хотим сделать FFT по модулю простого числа?

Заметим, что в изначальном алгоритме нам нужен лишь корень степени $n = 2^t$ из единицы. В остатках по модулю простого числа вида $p = 2^t \cdot c + 1$ такие числа тоже встречаются. Несколько частых модулей для NTT:

- $998244353 = 119 \cdot 2^{23} + 1$

  Первообразный корень: 3

  Соответственно, первообразный корень из единицы 2^{23}-й степени: 3^{119}

- $786433 = 3 \cdot 2^{18} + 1$

  Первообразный корень: 10

- ~~Другие не попадаются~~

Если вдруг попался модуль, для которого вы не знаете первообразного корня, то просто начните перебирать все числа с единицы подряд (g первообразный тогда и только тогда, когда $g^{\frac{p-1}{q}} \not\equiv 1 \pmod{p}$ для всех простых $q \mid p-1$). В предположении истинности обобщенной гипотезы Римана наименьший первообразный корень есть $O(\log^6 p)$.