Многочлены 101

Тимофей Равнушкин, Иван Пискарёв

Январь 2026

Содержание

1 Нотация

Все многочлены и степенные ряды в этом конспекте рассматриваются над вещественными числами, если не указано иное.

Определение 1.0.1

Формальным степенным рядом от переменной x (ФСР) называется бесконечная сумма $A(x) = a_0 + a_1x + a_2x^2 + \dots$ (вне зависимости от того, сходится такой ряд при каких-либо значениях x или нет). На формальных степенных рядах определено умножение на вещественное число (поэлементное), сумма (поэлементная) и произведение, эквивалентное произведению многочленов (свёртка последовательностей коэффициентов):

$$A(x)B(x) = (\sum_{i=0}^{\infty} a_ix^i)(\sum_{j=0}^{\infty} b_jx^j) = (\sum_{k=0}^{\infty} \sum_{i+j=k} a_ib_jx^k) = C(x)$$

Будем обозначать коэффициент при x^n в $A(x)$ за $[x^n]\, A(x)$.

Определение 1.0.2

Если A, B - ФСР и $B(0) = 0$, либо A - многочлен, то можно определить **композицию** $A(B(x)) = \sum_{i=0}^{\infty} a_i B^i(x)$.

Экспонентой называется ФСР $\exp(x) = \sum\limits_{i=0}^{\infty} \frac{x^i}{i!}$.

Экспонентой степенного ряда A называется композиция $\exp(A)$.

Логарифмом степенного ряда A называется такой ФСР B, что $\exp(B) = A$.

Определение 1.0.3

Формальной производной ФСР $A(x) = a_0 + a_1 x + \dots$ называется ФСР $A'(x) = a_1 + 2a_2 x + \dots = \sum\limits_{i=0}^{\infty} (i+1) a_{i+1} x^i$

Определение 1.0.4

Общеизвестно, что для любых многочленов A и B существует единственное разложение

$$A = DB + R, \deg R < \deg B$$

Будем называть R **остатком** A **по модулю** B.

Многочлены A и B называются сравнимыми по модулю C, если у них одинаковые остатки по модулю C. (Запись: $A \equiv B \pmod{C}$)

Для формального степенного ряда $A(x) = \sum_0^{\infty} a_i x^i$ будем называть его остатком по модулю x^{n+1} многочлен $a_0 + a_1 x + \dots + a_n x^n$.

2 Операции с многочленами и рядами

2.1 Обращение степенного ряда

Для любого ФСР $A(x)$ такого, что $A(0) \neq 0$, существует ФСР $A^{-1}(x)$ такой, что $A^{-1}A = 1$. Давайте научимся находить его первые n членов (будем считать, что мы умеем получать коэффициенты A за $O(1)$).

Начнём с $B_0 = a_0^{-1} \equiv A^{-1} \pmod{x}$ и будем удваивать степень x, по модулю которой у нас есть обратный (т. е. удваивать количество найденных коэффициентов).

Пусть $B_k \equiv A^{-1} \pmod{x^a}$, где $a = 2^k$. Тогда для следующего B_{k+1} должно быть выполнено $AB_{k+1} \equiv 1 \pmod{x^{2a}}$:

$$\begin{gathered}
AB_k \equiv 1 \pmod{x^a} \\
1 - AB_k \equiv 0 \pmod{x^a} \\
1 - 2AB_k + A^2 B_k^2 \equiv 0 \pmod{x^{2a}} \\
1 \equiv A(2B_k - AB_k^2) \pmod{x^{2a}} \\
B_{k+1} = B_k(2 - AB_k)
\end{gathered}$$

Таким образом мы удваиваем количество известных нам коэффициентов за 2 умножения многочленов. Итоговая асимптотика $T(n) = T(n/2) + O(n \log n) = O(n \log n)$

2.2 Деление с остатком

Пусть у нас есть многочлены $A(x)$ и $B(x)$ степеней n и m, соответственно, причем $n > m$. Мы хотим найти такие многочлены $Q(x)$, $R(x)$, что

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

Определение 2.2.1

Реверсированным многочленом многочлена $P(x)$ называется многочлен $rev(P(x)) = x^{\deg P(x)}P(x^{-1})$. (На самом деле, это просто формализация разворота коэффициентов.)

В реверсированных многочленах можно записать

$$\begin{aligned} A(x) - R(x) &= B(x)Q(x) \\ rev(A(x)) - x^{n-\deg R}rev(R(x)) &= rev(B(x))rev(Q(x)) \\ rev(A(x)) &\equiv rev(B(x))rev(Q(x)) \pmod{x^{n-m+1}} \\ rev(Q(x)) &\equiv rev(A(x))rev(B(x))^{-1} \pmod{x^{n-m+1}} \end{aligned}$$

2.3 Метод Ньютона

Пусть у нас есть функция $F(x) = \sum_{i=0}^{\infty} \alpha_i(x-\beta)^i$, где $\alpha_i, \beta \in \mathbb{Z}[x]$. Мы хотим найти такой ФСР $P(x)$, что $F(P(x)) = 0$. Однако, мы не умеем хранить ФСР целиком, поэтому, как и в случае с обращением рядов, будем вычислять $P(x) \pmod{x^{2^k}}$ для разных k.

Предложение 2.3.1

Для любой функции F, имеющей вид из начала абзаца, верно разложение

$$F(x+y) = F(x) + F'(x)y + G(x,y)y^2$$

где y - формальная переменная, а $G(x,y)$ - какой-то ряд.

Доказательство. Пользуясь биномом Ньютона и изолируя первые 2 члена из него, получим

$$\begin{aligned} F(x+y) &= \sum_{i=0}^{\infty} \alpha_i(x-\beta+y)^i \\ &= \alpha_0 + \sum_{i=1}^{\infty}(\alpha_i((x-\beta)^i + i(x-\beta)^{i-1}y) + g_i(x,y)y^2) = \\ &= \sum_{i=0}^{\infty} \alpha_i(x-\beta)^i + \sum_{i=1}^{\infty} \alpha_i i(x-\beta)^{i-1}y + \sum_{i=0}^{\infty} g_i(x,y)y^2 = \\ &= F(x) + F'(x)y + G(x,y)y^2 \end{aligned}$$

□

Пусть $F(Q_k) \equiv 0 \pmod{x^a}$. Мы хотим найти $Q_{k+1} \equiv Q_k + x^aC \pmod{x^{2a}}$ такой, что $F(Q_{k+1}) \equiv 0 \pmod{x^{2a}}$. Подставляя $x = Q_k, y = x^aC$ в разложении из 2.3.1, получим

$$F(Q_{k+1}) = F(Q_k) + F'(Q_k)x^aC + G(Q_k, x^aC)x^{2a}C^2$$

Пользуясь $x^aC \equiv Q_{k+1} - Q_k \pmod{x^{2a}}$:

$$0 \equiv F(Q_{k+1}) \equiv F(Q_k) + F'(Q_K)(Q_{k+1} - Q_k) \pmod{x^{2a}}$$

$$\boxed{Q_{k+1} \equiv Q_k - \frac{F(Q_k)}{F'(Q_k)} \pmod{x^{2a}}}$$

Асимптотика $T(n) = T(n/2) + O($вычисление $F(Q_k)$ и $F'(Q_k)^{-1})$.

Заметим, что мы можем получить алгоритм для обращения многочлена из метода Ньютона, пользуясь $F(x) = x^{-1} - P$, и формула будет в точности эквивалентной 2.1.

2.4 Формальная производная

Пусть $F(x) = \sum\limits_{n=0}^{\infty} f_n x^n, G(x) = \sum\limits_{n=0}^{\infty} g_n x^n$. Так как

$$(x^n \cdot x^m)' = (n+m)x^{n+m-1} = (x^n)' \cdot x^m + (x^m)' \cdot x^n$$

То $(F(x)G(x))' = F'(x)G(x) + G'(x)F(x)$ В частности $((F(x))^n)' = n \cdot F'(x) \cdot (F(x))^{n-1}$.

Значит

$$(F(G(x)))' = \left(\sum_{n=0}^{\infty} f_n (G(x))^n\right)' = \sum_{n=0}^{\infty} f_n \cdot n \cdot G'(x) \cdot (G(x))^{n-1} = F'(G(x)) \cdot G'(x)$$

Определим

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$
$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^n}{n}$$

Тогда

$$\exp'(x) = \exp(x)$$
$$\ln'(1+x) = \sum_{n=1}^{\infty} (-1)^{n-1} x^n = \frac{1}{1+x} \implies \ln'(F(x)) = \frac{1}{F(x)}$$

Заметим, что

$$(\ln(F(x)))' = \ln'(F(x))F'(x) = \frac{F'(x)}{F(x)}$$
$$(\ln(\exp(F(x))))' = \ln'(\exp(F(x))) \cdot \exp'(F(x)) \cdot F'(x) = \frac{\exp'(F(x))}{\exp(F(x))} \cdot F'(x) = F'(x)$$
$$\implies \ln(\exp(F(x))) = F(x)$$

2.5 Логарифм

Для функции $\ln P(x)$ известно, что $(\ln P(x))' = \dfrac{P'(x)}{P(x)}$. Мы можем вычислить P' за линию, найти обратный к нему за $O(n \log n)$, перемножить с P за $O(n \log n)$ и затем посдвигать коэффициенты за линию, чтобы восстановить ln по его производной.

2.6 Экспонента

Для $\exp(P) = Q$ верно, что $\ln Q = P$, поэтому воспользуемся методом Ньютона для $F(x) = \ln x - P$:

$$F(x) = \ln x - P$$
$$F'(x) = x^{-1}$$
$$\boxed{Q_{k+1} \equiv Q_k(1 + P - \ln Q_k) \pmod{x^{2^{k+1}}}}$$

Для возведения многочлена в k-ю степень вспомним, что $P^k = \exp(k \ln P)$.

3 Вычисление и интерполяция

3.1 Chirp-z transform

Пусть нам дан многочлен $P(x) = \sum_{i=0}^{n-1} a_i x^i$ и числа c и m, и мы хотим вычислить $P(c^0), P(c^1), \ldots, P(c^m)$.

Обозначим $b_j = P(c^j) = \sum_{i=0}^{n-1} a_i c^{ij}$. Заметим, что $ij = \frac{(i+j)^2 - i^2 - j^2}{2}$. Получается, что:

$$b_j c^{\frac{j^2}{2}} = \sum_{i=0}^{n-1} \left(a_i c^{-\frac{i^2}{2}}\right) c^{\left(\frac{(i+j)^2}{2}\right)}$$

В такой форме b_j можно найти свёрткой $a_i c^{-\frac{i^2}{2}}$ и $c^{\frac{i^2}{2}}$. Однако, во многих случаях найти $c^{\frac{i^2}{2}}$ невозможно (например, часто такое бывает при работе с остатками по модулю). Как же быть?

Несложными алгебраическими манипуляциями можно получить, что $ij = \binom{i+j}{2} - \binom{i}{2} - \binom{j}{2}$. Аналогично предыдущему случаю, получим, что:

$$b_j c^{\binom{j}{2}} = \sum_{i=0}^{n-1} (a_i c^{-\binom{i}{2}}) c^{\binom{i+j}{2}}$$

$$b_j c^{\binom{j}{2}} = \sum_{i=0}^{n-1} (a_{n-(n-i)} c^{-\binom{n-(n-i)}{2}}) c^{\binom{i+j}{2}}$$

Если ввести $C(x) = \sum_{i=1}^{n} a_{n-i} c^{-\binom{n-i}{2}} x^i$, $D(x) = \sum_{i=0}^{\infty} c^{\binom{i}{2}} x^i$, то получим, что:

$$\boxed{P(c^j) = c^{-\binom{j}{2}} \left[x^{n+j}\right] (C(x) \cdot D(x))}$$

3.2 Multipoint evaluation

Пусть нам дан многочлен $A(x)$ и набор чисел $x_1, \ldots, x_m$ и мы хотим вычислить значения $A(x_1), \ldots, A(x_m)$. Очевидно, что $A(x_i) \equiv A(x) \pmod{x - x_i}$. При этом известно, что если $A(x) \equiv B(x) \pmod{C \cdot D}$, то $A(x) \equiv B(x) \pmod{C}$.

Обозначим $P_{l,r}(x) = \prod_{i=l}^{r}(x - x_i)$. Мы можем сделать разделяйку: пусть у нас уже вычислено значение $A(x) \pmod{P_{l,r}(x)}$, тогда можно запустить следующие итерации с $A(x) \pmod{P_{l,m}(x)}$ и $A(x) \pmod{P_{m+1,r}(x)}$, тем самым уменьшив степень остатков $A(x)$ вдвое. Если заранее посчитать нужные $P_{l,r}(x)$, алгоритм суммарно будет работать за $O(n \log^2 n)$ (у нас $\log n$ слоев разделяйки, на каждом суммарно n коэффициентов).

3.3 Интерполяция через мультипоинт

Пусть нам дан набор из n пар (x_i, y_i) и мы хотим найти многочлен $A(x)$ такой, что для всех i от 1 до n $A(x_i) = y_i$. Это можно сделать, решив СЛАУ с коэффициентами многочлена, однако, это долго. Известна формула интерполяции Лагранжа:

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

Очевидно, что при подстановке $x = x_i$ в 0 обратятся все члены суммы, кроме члена с y_i.

Как вычислить коэффициенты $A(x)$ за быстро?

Рассмотрим многочлен $P(x) = \prod_{i=1}^{n}(x - x_i)$. Если посмотреть на его производную $P'(x)$, можно заметить, что если подставить в неё x_i, получится в точности $\prod_{j \neq i}(x_i - x_j)$, коэффициент, стоящий в знаменателе у y_i.

Получается, что мы свели задачу к вычислению $\sum_{i=1}^{n} a_i \prod_{j \neq i}(x - x_j)$, что можно сделать разделяйкой вида $A_{l,r} = A_{l,m} P_{m+1,r} + A_{m+1,r} P_{l,m}$. Итого мы умеем интерполировать многочлен за $O(n \log^2 n)$.

4 Half-GCD

Нам даны многочлены $A(x), B(x)$ и мы хотим вычислить многочлен максимальной степени $P(x)$ такой, что $P(x) \mid A(x), B(x)$. Будем делать это рекурсивно (классический алгоритм Евклида работает за $O(\deg A \deg B)$, а мы хотим научиться за $O(n \log^2 n)$).
Заметим, что шаг стандартного Евклида можно записать, как умножение столбца из 2 многочленов на матрицу – если $A(x) = B(x)Q(x) + R(x)$, то

$$\begin{pmatrix} B(x) \\ R(x) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -Q(x) \end{pmatrix} \begin{pmatrix} A(x) \\ B(x) \end{pmatrix}$$

И весь алгоритм тогда можно записать, как последовательность умножений на матрицы, то есть умножение на их произведение!

Заметим, что если у нас будет функция $HGCD(A(x), B(x))$, которая по паре многочленов будет за $O(n \log^2 n)$ возвращать матрицу, умножение на которую уменьшает степень второго аргумента хотя бы вдвое, то весь алгоритм можно будет свести к двум шагам:

1. $M = HGCD(A(x), B(x))$
2. $\begin{pmatrix} C(x) \\ D(x) \end{pmatrix} = M \begin{pmatrix} A(x) \\ B(x) \end{pmatrix}$
3. поделить $C(x)$ на $D(x)$ с остатком и продолжить с $D(x)$ и остатком в качестве многочлена

В итоге это будет работать за $O(n \log n)$, потому что степени обоих многочленов после шага уменьшатся вдвое. Осталось научиться считать
$HGCD(A(x), B(x))$.

Заметим, что для шагов алгоритма Евклида, в результате композиции которых степень меньшего из многочленов уменьшается хотя бы на d, достаточно знать старшие $2d$ коэффициентов обоих многочленов. Из этого придумывается способ считать $HGCD$:

Пусть степень $A(x)$ равна $4d$, а степень $B(x)$ меньше степени $A(x)$. Для того, чтобы убрать старшие $\frac{4d}{2} = 2d$ коэффициентов, два раза уберём по d. Но для того, чтобы убрать d, нам достаточно знать только старшие $2d$ коэффициентов! Поэтому можно запустить $M = HGCD(A(x) \,\mathbf{div}\, x^{2d}, B(x) \,\mathbf{div}\, x^{2d})$, затем полученную матрицу применить к $\begin{pmatrix} C(x) \\ D(x) \end{pmatrix} = M \begin{pmatrix} A(x) \\ B(x) \end{pmatrix}$, поделить $C(x)$ на $D(x)$ с остатком (это эквивалентно матрице L, полученной, как описано ранее) и от старших коэффициентов результата ещё раз запустить $HGCD$, получить матрицу K и вернуть KLM.

5 Поиск корней над $\mathbb{Z}/p\mathbb{Z}$

Пусть нам дан многочлен $q(x)$ с целыми коэффициентами, мы хотим найти все такие $a \in \mathbb{Z}/p\mathbb{Z}$, что $q(a) \equiv 0 \pmod p$.

Для начала заменим наш многочлен на $h(x) = \gcd(q(x), x^p - x)$, чтобы избавиться от нелинейных или повторяющихся делителей в разложении q (так как $x^p - x \equiv \prod_{a=0}^{p-1}(x-a) \pmod p$).

После этого выберем случайное $a \in \mathbb{Z}/p\mathbb{Z}$ и вычислим

$$f(x) = \gcd(h(x), (x+a)^{\frac{p-1}{2}} - 1)$$
$$g(x) = \frac{h(x)}{f(x)}$$

И будем решать задачу рекуррентно для f и g.

В $f(x)$ останутся такие корни λ, что $h(\lambda) = 0$ и $(\lambda + a)$ является квадратичным вычетом по модулю p. Почему это работает быстро? Квадратичных вычетов всего $\frac{p-1}{2}$ и они распределены равномерно, поэтому вероятность попасть в такой при прибавлении a для каждого из корней $h(x)$ примерно равна $\frac{1}{2}$. Получается, что вероятность получить $\deg f = k < n = \deg h$ приблизительно $2^{-n}\binom{n}{k}$, поэтому на каждом шаге степень понижается почти вдвое. Итого асимптотика $O(n \log n \log p \log(np))$, если перемножать всё с помощью FFT и использовать half-GCD.

6 Taylor Shift

Нам дан многочлен $P(x)$, мы хотим найти коэффициенты $P(x+a)$.

Заметим, что

$$(x+a)^k = \sum_{i=0}^{k} \binom{k}{i} a^i x^{k-i} = k! \sum_{i=0}^{k} \frac{a^i}{i!} \frac{x^{k-i}}{(k-i)!}$$

$$\frac{(x+a)^k}{k!} = \sum_{i+j=k} \frac{a^i}{i!} \frac{x^j}{j!}$$

Если обозначить оператор дифференцирования за $D = \frac{\mathrm{d}}{\mathrm{d}x}$, то $D\frac{x^k}{k!} = \frac{x^{k-1}}{(k-1)!}$. Тогда наше выражение можно переписать, как

$$\frac{(x+a)^k}{k!} = \sum_{i=0}^{k} \frac{a^i}{i!} \left(D^i \frac{x^k}{k!} \right) = \left(\sum_{i=0}^{\infty} \frac{a^i D^i}{i!} \right) \frac{x^k}{k!} = \exp(aD) \frac{x^k}{k!}$$

Тут $\exp(aD)$ - экспонента оператора aD, формально определённая, как $\sum_{i=0}^{\infty} \frac{(aD)^i}{i!}$.

Дифференцирование коммутирует с умножением на скаляр, поэтому $(aD)^i = a^i D^i$, а ещё мы можем избавиться от $k!$ и получить $(x+a)^k = \exp(aD)x^k$. Дифференцирование линейно, поэтому $\exp(aD)P(x) = P(x+a)$ для любого многочлена $P(x)$.

Как научиться применять ряд от D к многочлену?

Введём линейные операторы $[\cdot]$ и $\{\cdot\}$ на пространстве многочленов, действующие на мономах, как $[x^k] = \frac{x^k}{k!}$ и $\{x^k\} = k!x^k$.

Заметим, что во-первых $[\{P(x)\}] = \{[P(x)]\} = P(x)$ для любого многочлена $P(x)$.

Во-вторых,

$$D^i[x^j] = \frac{x^{j-i}}{(j-i)!} = [x^{j-i}]$$

что можно переписать, как

$$D^i[P(x)] = [x^{-i}P(x)]$$

для любого многочлена $P(x)$. (Тут мы определяем $n! = \infty$ для $n < 0$, поэтому $[x^k]$ равно нулю для отрицательных k).

Благодаря линейности $[\cdot]$, второе замечание можно расширить до

$$G(D)[P(x)] = [G(x^{-1})P(x)]$$

для любого многочлена G. Совмещая это с первым замечанием, получим, что

$$G(D)P(x) = [G(x^{-1})\{P(x)\}]$$

Итого получаем, что

$$P(x+a) = \left[\exp\left(ax^{-1}\right)\{P(x)\}\right]$$

что можно вычислить за $O(n \log n)$.

7 Базис нисходящих факториалов

7.1 Определения

Определение 7.1.1

k**-м нисходящим факториалом** называется многочлен $(x)_k = x(x-1)\dots(x-(k-1))$

Будем считать $(x)_0 = 1$

Степень k-го нисходящего факториала равна k, поэтому они образуют базис векторного пространства многочленов (будем называть его биномиальным базисом) (стандартный базис – мономиальный – это $1, x, x^2, \dots$).

Определение 7.1.2

Дискретная производная Δ функции f определяется, как $\Delta f(x) = f(x+1) - f(x)$.

Заметим, что в терминах нисходящих факториалов $\binom{n}{k} = \frac{(n)_k}{k!}$. Пользуясь $\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$, получаем, что $\Delta\binom{x}{k} = \binom{x}{k-1}$. Следовательно,

$$\Delta(x)_k = k!\Delta\binom{x}{k} = k!\binom{x}{k-1} = k(x)_{k-1}$$

То есть дискретная производная в биномиальном базисе действует на многочлены так же, как обычная - в мономиальном! В этом базисе можно частично обратить Δ (полностью этот оператор необратим, так как $\Delta 1 = 0$) – если мы знаем, что $\Delta p(x) = (x)_k$, то $p(x) = \frac{1}{k+1}(x)_{k+1} + C$ для какой-то константы C. Будем называть эту операцию псевдоинтегрированием.

7.2 Простое применение

Рассмотрим простое случайное блуждание на $1, \ldots, n$ - процесс, при котором мы начинаем в 1 и каждый шаг происходит следующее:

1. если мы сейчас в 1, мы переходим в 2
2. если мы сейчас в $x \neq 1$, мы переходим в $x-1$ с вероятностью $\frac{1}{2}$ и в $x+1$ с вероятностью $\frac{1}{2}$

Пусть мы хотим найти матожидание количества шагов до того, как мы попадём в n.

Обозначим за E_x матожидание количества шагов до n, если мы начинаем в x. Тогда $E_1 = 1 + E_2$, $E_n = 0$, и

$$E_x = 1 + \frac{1}{2}E_{x-1} + \frac{1}{2}E_{x+1}$$

для $1 < x < n$. Заметим, что

$$\begin{aligned}
\Delta^2 E_x &= (\Delta E)_{x+1} - (\Delta E)_x \\
\Delta^2 E_x &= E_{x+2} - E_{x+1} - E_{x+1} + E_x \\
\Delta^2 E_x &= E_{x+2} - 2E_{x+1} + E_x \\
\Delta^2 E_x &= -2 \\
\Delta E_1 &= -1 \\
E_n &= 0
\end{aligned}$$

Псевдоинтегрируя $\Delta^2 E_x = -2$, получим, что $\Delta E_x = -2(x)_1 + C = -2x + C$ для некоторой константы C. Второе уравнение даёт $C = 1$. Псевдоинтегрируя снова, получим $E_x = -(x)_2 + x + C_1 = -x^2 + 2x + C_1$ для некоторой C_1, которая из третьего уравнения равна $n^2 - 2n$.

7.3 Вычисление, интерполяция и смена базиса

Если мы научимся вычислять и интерполировать многочлены, записанные в биномиальном базисе, мы получим способ менять базис с мономиального на биномиальный и наоборот, потому что вычислять и интерполировать в нём мы уже умеем. Начнём с вычисления. Пусть нам дан многочлен $P(x) = \sum\limits_k \alpha_k (x)_k$ и мы хотим вычислить $P(0), P(1), \ldots, P(d)$. Для точки m

$$\begin{aligned}
P(m) &= \sum_k \alpha_k (m)_k \\
&= \sum_{k=0}^{m} \alpha_k (m)_k \\
&= \sum_{k=0}^{m} \alpha_k \frac{m!}{(m-k)!},
\end{aligned}$$

то есть

$$\frac{P(m)}{m!} = \sum_{k=0}^{m} \alpha_k \frac{1}{(m-k)!}$$

Правая сторона этого выражения - свёртка, поэтому можно переписать

$$\sum_m P(m)\frac{x^m}{m!} = \left(\sum_k \frac{x^k}{k!}\right)\left(\sum_k \alpha_k x^k\right) = \exp(x)\left(\sum_k \alpha_k x^k\right)$$

Интерполяция настолько же проста - домножим обе стороны на $\exp(-x)$ и получим выражение для коэффициентов α_k из значений $P(m)$. Поэтому в биномиальном базисе можно вычислять на отрезке точек и интерполировать из него за $O(n \log n)$.

7.4 Сдвиги

Пусть у нас есть многочлен $P(x)$, записанный в биномиальном базисе, и мы хотим вычислить коэффициенты $P(x+c)$. Заметим, что

$$P(x+1) = P(x) + \Delta P(x) = (1+\Delta)P(x)$$

Следовательно, для целого c верно

$$P(x+c) = (1+\Delta)^c P(x) = \sum_{k=0}^{c} \binom{c}{k} \Delta^k P(x)$$

Если степень P равна d, то нам нужны только первые $d+1$ коэффициентов выражения выше. Обе стороны - многочлены от c, поэтому мы можем вычислять сдвиги и для нецелых значений c (пользуясь обобщённым биномом Ньютона).

Мы уже умеем применять ряд от D к многочлену в мономиальном базисе, поэтому можно временно “забыть” о том, что мы работаем в биномиальном, заменив $P(x) = \sum_{i=0}^{n} a_i (x)_i$ на $A(x) = \sum_{i=0}^{n} a_i x^i$ и поменяв везде Δ на D, потому что на соответствующих базисах они действуют одинаково.