1 Быстрое преобразование Фурье

1.1 Дискретное преобразование Фурье в поле комплексных чисел

Рассмотрим некоторый многочлен $A = a_0 + a_1 \cdot x + \cdots + a_{n-1} \cdot x^{n-1}$ над полем C. Здесь и далее степенью многочлена будет считаться число коэффициентов в нем.

Пусть $\omega = e^{\frac{2\pi i}{n}}$ — корень n-ой степени из 1.

Дискретным преобразованием Фурье многочлена A называется вектор, состоящий из n значений, определяемый как

$$DFT(A)_k = A(\omega^k) = \sum_{j=0}^{n-1} a_j \cdot e^{\frac{2\pi ijk}{n}}, k = 0..n-1$$

Это можно рассматривать, как отображение $DFT : \mathbb{C}^n \to \mathbb{C}^n$. Докажем, что такое отбораженение обратимо.

Рассмотрим преобразование DFT^{-1}, которое последовательности $B = \langle b_0, b_1, \cdots b_{n-1} \rangle$ сопостовляет вектор из n комплексных чисел, определяемое по формуле

$$DFT^{-1}(B)_k = \frac{1}{n} \sum_{j=0}^{n-1} a_j \cdot e^{-\frac{2\pi ijk}{n}}, k = 0..n-1$$

Взаимно-обратность этих отображений доказывается явной подстановкой. Несложно заметить, что обратное преобразование Фурье, получается из обычного, выбором в качестве корня ω^{-1}. Для большей симметричности, иногда вместо сомножетеля $\frac{1}{n}$ у обратного преобразования, пишут сомножетель $\frac{1}{\sqrt{n}}$ у обоих, однако для наших целей такой вариант нормировки будет менее удобен.

Таким образом у нас есть взаимно-однозанчное состествие многочленов степени не более n, с последовательностями из n целых чисел.

Полезным в этой биекции является ее связь с перемножением многочленов. Из определения очевидно, что $DFT(A \cdot B) = DFT(A) \cdot DFT(B)$. И если многочлен $A \cdot B$ так же степени не более n, то применив обратное преобразование Фурье к обеим частям равенства, $A \cdot B = DFT^{-1}(DFT(A) \cdot DFT(B))$[1]. Оказывается, что задача вычисления преобразования Фурье вычислительно проще, чем тривиальное премножение многочленов.

[1] В общем случае, произведение будет вычислено в поле $\mathbb{C}[x]/(x^n - 1)$

1.2 Быстрое преобразование Фурье

Не умаляя общности, можно считать что $n = 2^m$. Для этого необходимо его увеличить не более, чем в два раза.

Запишем наш многочлен в следующем виде

$$A(x) = A_1(x^2) + xA_2(x^2)$$

. С точки зрения последовательностей, эта операция выглядит, как разбиение на четные и нечетные позиции.

Вычислим рекурсивно преобразования Фурье для многочленов A_1 и A_2. Заметим, что полученных данных достаточно для вычисления $DFT(A)$ по формулам:

$$DFT(A)_k = DFT(A_1)_k + \omega^k DFT(A_2)_k, k = 0..\frac{n}{2} - 1$$

$$DFT(A)_{k+\frac{n}{2}} = DFT(A_1)_k - \omega^k DFT(A_2)_k, k = 0..\frac{n}{2} - 1$$

Эта пара формул часто называется преобразованием бабочки.

Время работы это алгоритма вычисляется из рекурентного соотношения $T(n) = 2T(\frac{n}{2}) + O(n)$ и равно $O(n \log n)$

1.3 Нерекурсивная реализация

Базовый алгоритм, описанный выше, имеет достаточно большую константу скрытую под знаком O-большое. Это связано с большим количеством рекурсивных вызовов, а так же неоптимальным перемещением по памяти.

Рассмотрим нерекурсивную реализацию этого алгоритма. Фактически, алгоритм сначала переупорядычивает элементы в правильном порядке, потом объединяет их группами по 2, потом по 4 и так далее.

Для начала, необходимо понять, что это за порядок. Несложно проверить, что порядок соответсвует перемещению числа с позиции i на позицию $bit\_reverse(i)$[2]. Такую перестановку можно выполнить за линию, заранее предподсчитав $bit\_reverse$.

После чего необходимо $m = \log n$ раз склеить соответсвующие части массива в одну. Формулы вычисления для этого очень удобны. Несложно заметить, что если записывать результат склейки на место исходных двух массивов (они шли подряд), то элементы $DFT(A)_k$ и $DFT(A)_{k+\frac{n}{2}}$

[2] $bit\_reverse(i)$ — функция, сопостовляющая числу i, число двоичная запись которого дополненная до m знаков нулями, является перевернутой двоичной записью числа i

надо поставить ровно на место тех двух элементов по которым они вычисляются. Таким образом на k-ой итерации необходимо просто применить преобразование бабочки ко всем парам чисел на расстоянии 2^k.

1.4 Прочие технические оптимизации

Одной из самых долгих частей является постоянное вычисление корней из 1. Поэтому из лучше предподсчитать.

Кроме того, часто коэффициенты исходных многочлена вещественные. Научимся сводить вычисление преобразований Фурье двух вещественных многочленов к вычислению преобразования одного комплексного.

Пусть P, Q — вещественные многочлены. Пусть $S = P + iQ$

Попытаемся выразить $DFT(P), DFT(Q)$ через $DFT(S)$.

$$P = \frac{(P + iQ) + (P - iQ)}{2} = \frac{S + \overline{S}}{2}$$
$$Q = \frac{(P + iQ) - (P - iQ)}{2i} = \frac{S - \overline{S}}{2i}$$

Тогда

$$DFT(P)_k = P(\omega^k) = \frac{S(\omega^k) + \overline{S}(\omega^k)}{2} = \frac{S(\omega^k) + \overline{S(\omega^{n-k})}}{2} = \frac{DFT(S)_k + \overline{DFT(S)_{n-k}}}{2}$$

$$DFT(Q)_k = Q(\omega^k) = \frac{S(\omega^k) - \overline{S}(\omega^k)}{2i} = \frac{S(\omega^k) - \overline{S(\omega^{n-k})}}{2i} = \frac{DFT(S)_k - \overline{DFT(S)_{n-k}}}{2i}$$

1.5 Преобразование Фурье в поле остатоков по модулю p

Пусть p — простое число. В таком случае, тот же алгоритм применим к многочленам по модулю p, если в качестве ω взять $r^{\frac{p-1}{n}}$, где r — первообразный корень. Для этого необходимо, чтобы $p-1$ делилось на большую степень двойки.

Единственным вопросом остается как эффективно найти первообразный корень. Для этого воспользуемся тем, что их много, и просто будем выбирать случайный элемент, пока не наткнемся на первообразный корень. Для проверки, что r первообразный корень, необходимо проверить, что для всех простых делителей q числа $p-1$, $r^{\frac{p-1}{q}} \not\equiv 1(modp)$. Такую проверку можно осущствить за время $\log^2(p)$, если заранее предподсчитать простые делители, а так же воспользоваться быстрым возведением в степень. Простые делители можно найти за время $O(\sqrt{p})$ (а

на самом деле и быстрее). Первообразных корней хотя бы $\phi(n) \geq \frac{n}{\ln n}$. То есть математическое ожидание времени работы такого алгоритма $O(\log^3 n + \sqrt{n})$.

1.6 Примеры

Перемножение длинных чисел. Представим длинное число как многочлен от базы.

Задачи поиска подстроки с наименьшим числом ошибок.

Задача об циклическом сдвиге с минимальным скалярным произведением

1.7 Обращение формального степенного ряда

Часто на производящие функции многих величин можно написать рекурентные соотношения.

В которых к производящим функциям приходится применять такие операции как обращение.

Научится вычислять $P^{-1} \mod x^n$, как обычно будем считать, что $n = 2^m$.

Пусть $PQ \equiv 1 \mod x^n$ Запишем $P(x) = P_1(x) + x^{\frac{n}{2}} \cdot P_2(x)$, $P(x) = Q_1(x) + x^{\frac{n}{2}} \cdot Q_2(x)$.

В таком случае $P_1 Q_1 \equiv 1 \mod x^{\frac{n}{2}}$. Вычислим его рекурсивно.

Теперь верно равенство:

$$P_2 Q_1 + Q_2 P_1 = 0(\mod x^{\frac{n}{2}})$$

. Домножим его на Q_1:

$$Q_2 = -P_2 Q_1^2(\mod x^{\frac{n}{2}})$$

.

Величину $P_2 Q_1^2$ можно вычислить за время $O(n \log n)$ с помощью преобразования Фурье.

Время работы такого алгоритма вычисляется из рекурентной последовательности $T(n) = T(n/2) + O(n \log n)$, решением которой является $O(n \log n)$

Чтобы избавиться от рекурсии, выразим Q, через Q_1 $Q = Q_1 - x^{\frac{n}{2}} P_2 Q_1^2 = 2Q_1 - Q_1^2(P_1 + x^{\frac{n}{2}} P_2) = 2Q_1 - Q_1^2 P(\mod x^n)$. Мы получили формулу, позволяющую удвоить число точных членов. Если каждый раз отбрасывать неотчные, время вычисления останется таким же. Стоит заметить, что эта формула совпадает с формулой метода Ньютона для уравнения $P - \frac{1}{X} = 0$.

Пример: разбиение числа на слагаемые. Производящая функция — $\frac{1}{\prod\limits_{k=1}^{\infty} 1-x^k}$. Известно, что $\prod\limits_{k=1}^{\infty} 1-x^k = \sum\limits_{q=-\infty}^{\infty} (-1)^q \cdot x^{\frac{3q^2+q}{2}}$. То, есть обратную производящую функцию по модулю x^n можно вычислить за время $O(n)$, после чего ее можно обратить за $O(n \log n)$. Как вычислить быстрее, чем динамикой за $O(n^3)$, без обращения степенных рядов — непонятно.