ИНФОРМАТИКА

Вестник Сыктывкарского университета.
Серия 1: Математика. Механика. Информатика.
Выпуск 4 (25). 2017

УДК 517.443, 519.688

ПРИМЕНЕНИЕ БПФ В ЗАДАЧАХ СПОРТИВНОГО ПРОГРАММИРОВАНИЯ

Н. О. Котелина

В этой статье рассматривается использование БПФ для решения одной задачи спортивного программирования.
Ключевые слова: дискретное преобразование Фурье, программирование.

1. Дискретное преобразование Фурье [2, 3]

Пусть имеется многочлен степени меньше n:

$$A(x) = a_0x^0 + a_1x^1 + \ldots + a_{n-1}x^{n-1}.$$

Будем считать, что n является степенью 2. Если в действительности это не так, то добавим недостающие коэффициенты, положив их равными нулю.

Обозначим за $\omega_{n,k} = e^{i\frac{2\pi k}{n}}$, $k = 0, \ldots, n-1$, комплексные корни n-й степени из 1. Очевидно, что все корни $\omega_{n,k}$ являются степенями главного значения корня n-й степени из единицы $\omega_n = \omega_{n,1}$:

$$\omega_{n,k} = \omega_n^k.$$

Дискретным преобразованием Фурье (ДПФ) многочлена $A(x)$ или, что то же самое, ДПФ вектора его коэффициентов $a = (a_0, a_1, \ldots, a_{n-1})$ называется вектор y значений этого многочлена в точках $x_k = \omega_n^k$, $k = 0, \ldots, n-1$:

$$y = \mathrm{DFT_n}(a) = (A(\omega_n^0), A(\omega_n^1), \ldots, A(\omega_n^{n-1})).$$

Можно определить и обратное дискретное преобразование Фурье DFT_n^{-1}. Обратным ДПФ для вектора y значений многочлена $A(x)$ в точках $x_k = \omega_n^k$, $k = 0, \ldots, n-1$, называется вектор его коэффициентов $a = (a_0, a_1, \ldots, a_{n-1})$:

$$a = \mathrm{DFT_n^{-1}}(y).$$

Известно, что ДПФ можно применять для вычисления коэффициентов произведения полиномов [3].

Пусть даны два многочлена $A(x)$ и $B(x)$ степени меньше n. Найдем вектор коэффициентов их произведения $C(x) = A(x)B(x)$. Очевидно, что полином $C(x)$ имеет степень меньше $2n-1$ (и тем более меньше $2n$). Пусть c — вектор его коэффициентов (включая нулевые) длины $2n$. Если дополнить векторы a и b нулевыми коэффициентами до длины $2n$, то элементы вектора c можно найти по формуле

$$c_k = \sum_{j=0}^{k} a_j b_{k-j}, \qquad k = 0, \ldots, 2n-1.$$

Вектор c называется свёрткой векторов a и b и обозначается $c = a \otimes b$.

Справедлива теорема о свёртке, аналогичная теореме 5.1 из книги [4]: Для любых векторов a и b размерности n, где n — степень 2, выполнено равенство

$$c = \mathrm{DFT_{2n}^{-1}}(\mathrm{DFT_{2n}}(a) \cdot \mathrm{DFT_{2n}}(b)),$$

если дополнить векторы a и b нулевыми элементами до длины $2n$ (точка здесь обозначает поэлементное произведение векторов). Для полноты изложения приведём краткое доказательство теоремы о свёртке.

Доказательство. Рассмотрим i-й элемент вектора $\mathrm{DFT_{2n}}(c)$:

$$\begin{aligned}
\mathrm{DFT_{2n}}(c)[i] &= \sum_{k=0}^{2n-1} c_k \omega_{2n}^{ik} = \sum_{k=0}^{2n-1} \Big(\sum_{j=0}^{k} a_j b_{k-j} \Big) \omega_{2n}^{ij} \omega_{2n}^{i(k-j)} = \\
&= \sum_{k=0}^{2n-1} \Big(\sum_{j=0}^{k} a_j \omega_{2n}^{ij} b_{k-j} \omega_{2n}^{i(k-j)} \Big) = \sum_{j=0}^{2n-1} a_j \omega_{2n}^{ij} \Big(\sum_{k=0}^{2n-1-j} b_k \omega_{2n}^{ik} \Big) = \\
&= \sum_{j=0}^{n-1} a_j \omega_{2n}^{ij} \Big(\sum_{k=0}^{2n-1-j} b_k \omega_{2n}^{ik} \Big) = \sum_{j=0}^{n-1} a_j \omega_{2n}^{ij} \Big(\sum_{k=0}^{n-1} b_k \omega_{2n}^{ik} \Big) = \\
&= \mathrm{DFT_{2n}}(a)[i] \cdot \mathrm{DFT_{2n}}(b)[i].
\end{aligned}$$

Отсюда следует искомая формула. □

2. Быстрое преобразование Фурье

Быстрое преобразование Фурье (fast Fourier transform) — это метод быстрого вычисления ДПФ за время $\Theta(n \log n)$, основанный на свойствах комплексных корней из единицы [2; 3, с. 723].

Быстрое преобразование Фурье использует метод «разделяй и властвуй», который заключается в разделении вектора коэффициентов на два вектора, рекурсивном вычислении ДПФ для них и объединении результатов в одно ДПФ. Для схемы «разделяй и властвуй» известна асимптотическая оценка $\Theta(n \log_2 n)$ [3]. Найти обратное ДПФ, т. е. по вектору значений y полинома $A(x)$ перейти к его вектору коэффициентов a также можно за время $\Theta(n \log_2 n)$, если применить тот же алгоритм БПФ, но с другими данными [3], поскольку для ДПФ и обратного ДПФ справедливы формулы

$$y_k = \sum_{j=0}^{n-1} a_j \omega_n^{kj}, \quad a_k = \frac{1}{n} \sum_{j=0}^{n-1} y_j \omega_n^{-kj}, \qquad k = 0, \dots, n-1.$$

Таким образом, по теореме о свёртке, вектор коэффициентов c произведения $A(x)B(x)$ может быть найден за время $\Theta(2n \log_2 2n) = \Theta(n \log_2 n)$.

3. Постановка задачи

Рассмотрим задачу «Вор в магазине» с ресурса [1], которая предлагалась на соревнованиях по спортивному программированию Educational Codeforces Round 9. Условие задачи таково.

Вор пробрался в магазин. Как всегда у него с собой любимый рюкзак. В рюкзаке может поместиться k предметов. В магазине присутствует n типов товаров, причём товаров каждого типа бесконечное количество. Стоимость единицы товара i-го типа равна a_i. Вор жадный, поэтому решил набить рюкзак до отказа. Таким образом, он возьмёт с собой ровно k товаров, причём товары некоторых типов он может взять в нескольких экземплярах. Определите всевозможные суммы стоимостей товаров, которые могут оказаться в рюкзаке вора.

Входные данные

В первой строке находится пара целых чисел n и k ($1 \leq n, k \leq 1000$) — количество типов товаров и количество предметов, которые вор украдёт. Во второй строке находятся n целых чисел a_i ($1 \leq a_i \leq 1000$) — стоимости товаров по типам от 1 до n.

Выходные данные

В единственной строке следует вывести через пробел всевозможные суммы стоимостей товаров, которые могут оказаться в рюкзаке вора. Числа нужно выводить в порядке возрастания.

Нетрудно видеть, что полный перебор всевозможных наборов товаров является неэффективным. Одно из эффективных решений данной задачи использует быстрое преобразование Фурье [1].

Обозначим за W максимально возможную стоимость украденных вором товаров. Тогда $W = k \max\{a_i,\, i \in 1:n\} \leq 10^6$.

Рассмотрим многочлен P степени $d = \max\{a_i,\, i \in 1:n\}$ с целочисленными коэффициентами, коэффициенты которого при степенях a_i, $i \in 1:n$, равны единице, а остальные равны нулю. Тогда при возведении P в степень k получим полином с ненулевыми коэффициентами при степенях $\{a_{i_1} + \cdots + a_{i_k} \mid 1 \leq i_j \leq n,\, j \in 1:k\}$, при этом коэффициент при степени val означает количество способов набрать k товаров с суммарной стоимостью val. Таким образом, эти степени и являются ответом на задачу. Если возводить полином в степень напрямую, то количество операций умножения будет равно $d^2 + 2d^2 + 3d^2 + \cdots + (k-1)d^2 = d^2 \dfrac{k(k-1)}{2}$ и при максимальных d и k может быть величиной порядка 10^{12}, что при существующих ограничениях на время работы программы слишком много. Покажем, что при помощи БПФ можно быстро возвести полином P в степень k так, чтобы общее число операций не превышало величину порядка 10^8.

4. Быстрое возведение полинома в степень

Как было замечено выше, при использовании быстрого преобразования Фурье асимптотическая сложность умножения двух полиномов степени n и, следовательно, возведения полинома степени n в квадрат равна $\Theta(n \log n)$. Пусть в нашей задаче максимальная суммарная стоимость товаров равна W, а $t = \lceil \log_2 W \rceil$, тогда по условию задачи $t \leq 20$. С другой стороны, W — это степень полинома, который получается в результате возведения в степень исходного полинома P. Тогда если дополнить исходный полином нулевыми коэффициентами до степени W, то время работы программы можно оценить следующим образом

$$(k-1)WlogW \leq 10^3 2^{20} 20 \approx 2 \cdot 10^{10}.$$

Эта величина все еще является слишком большой, поэтому, чтобы сократить количество умножений полиномов с $k-1$ до $\log_2 k$, можно воспользоваться алгоритмом бинарного возведения в степень [2; 3, с. 758]. Тогда общее число операций можно оценить так

$$WlogW \log_2 k \leq 10 \cdot 2^{20} 20 \approx 2 \cdot 10^8.$$

Данное количество операций удовлетворяет временным ограничениям задачи. На практике по времени проходит решение, использующее итеративную, а не рекурсивную реализацию БПФ, поскольку, несмотря на одинаковую асимптотическую сложность, итеративная реализация имеет меньшую константу [2; 3, с. 728].

Список литературы

1. Codeforces (c). Copyright 2010–2017. Михаил Мирзаянов. Соревнования по программированию 2.0. URL: http://codeforces.com. (дата обращения: 12.09.2017).

2. MAXimal. URL: http://e-maxx.ru. (дата обращения: 12.09.2017).

3. **Кормен Т., Лейзерсон Ч., Ривест Р.** Алгоритмы: построение и анализ. М.: МЦНМО, 2001. 960 с.

4. **Малозёмов В. Н., Машарский С. М.** Основы дискретного гармонического анализа. СПб.: Лань, 2012. 302 с.

Summary

Kotelina N. O. The application of FFT in problems of competitive programming

In this paper the use of FFT in problems of competitive programming is considered.

Keywords: discrete Fourier transform, competitive programming.

References

1. Codeforces (c). Copyright 2010–2017. Mihail Mirzayanov. *Sorevnovaniya po programmirovaniyu 2.0*: URL: http://codeforces.com. (date of the application: 12.09.2017).

2. *MAXimal*. URL: http://e-maxx.ru. (date of the application: 12.09.2017).

3. **Kormen T., Leiserson Ch., R. Rivest** *Algoritmy: postroeniye i analiz* (Algorithms: construction and analysis), M.: MCNMO, 2001, 960 p.

4. **Malozyomov V. N., Masharsky S. M.** *Osnovy diskretnogo garmonicheskogo analiza* (Fundamentals of discrete harmonic analysis), SPb.: Lan, 2012, 302 p.

Для цитирования: Котелина Н. О. Применение БПФ в задачах спортивного программирования // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2017. Вып. 4 (25). С. 44–49.*

For citation: Kotelina N. O. The application of FFT in problems of competitive programming, *Bulletin of Syktyvkar University. Series 1: Mathematics. Mechanics. Informatics*, 2017, №4 (25), pp. 44–49.

СГУ им. Питирима Сорокина *Поступила 20.11.2017*