жения, нам надо решить задачу *интерполяции*, т.е. восстановления коэффициентов многочлена по его значениям.

Интерполяционные полиномы

Каждый многочлен степени k однозначно определяется своими значениями в $k+1$ точке. В самом деле, пусть $P(x)$ и $Q(x)$ – два многочлена степени k, совпадающие в $k+1$ точке. Тогда их разность $P-Q$ обращается в ноль в этих точках и является многочленом степени не выше k. Но если многочлен степени не выше k имеет k нулей, то он тождественно равен нулю. Поэтому $P(x) \equiv Q(x)$. Однако нам нужно не просто доказать однозначность, а явно осуществить построение.

Построим многочлен k-й степени $R(x)$, принимающий в точках $x_1, \ldots, x_{k+1}$ значения $y_1, \ldots, y_{k+1}$. Для этого достаточно уметь строить элементарные многочлены $R_i(x)$, принимающие значения 1 в точке x_i и 0 в остальных точках. Тогда многочлен $R(x)$ находится как сумма

$$\sum_{i=1}^{k+1} y_i R_i(x).$$

Чтобы обеспечить равенство нулю в точках x_i, $i \neq j$, рассмотрим произведение

$$q_i(x) = (x-x_1)(x-x_2)\times\ldots \ldots\times\left(x-\hat{x}_i\right)\cdot\ldots\cdot(x-x_{k+1}),$$

где $\left(x-\hat{x}_i\right)$ означает, что соответствующий множитель опускается.

Разделив многочлен $q_i(x)$ на его значение в точке x_i, мы получим многочлен

$$R_i(x) = \frac{(x-x_1)(x-x_2)}{(x_i-x_1)(x_i-x_2)}\times\ldots \ldots\times\frac{\left(x-\hat{x}_i\right)\cdot\ldots\cdot(x-x_{k+1})}{\left(x_i-\hat{x}_i\right)\cdot\ldots\cdot(x_i-x_{k+1})}.$$

Окончательно имеем:

$$R(x) = \sum_{i=1}^{k+1} y_i\cdot\frac{(x-x_1)(x-x_2)}{(x_i-x_1)(x_i-x_2)}\times\ldots \ldots\times\frac{\left(x-\hat{x}_i\right)\cdot\ldots\cdot(x-x_{k+1})}{\left(x_i-\hat{x}_i\right)\cdot\ldots\cdot(x_i-x_{k+1})}.$$

Мы получили *интерполяционную формулу Лагранжа* для многочлена.

Найдем коэффициенты a_s многочлена R и коэффициенты a_{si} многочленов R_i. Воспользовавшись формулой Виета, получим

$$\sum_{j_1<\ldots<j_s;\, j_\alpha\neq i} x_{j_1}x_{j_2}\cdot\ldots\cdot x_{j_s}\Big/\big((x_i-x_1)\times$$

$$\times(x_i-x_2)\cdot\ldots\cdot(x_i-\hat{x}_i)\cdot\ldots\cdot(x_i-x_{k+1})\big).$$

Тогда

$$a_s = \sum_i y_i\cdot a_{si}.$$

Итак, мы получили формулы для коэффициентов интерполяционного полинома. Они быстро усложняются с ростом k – числа блоков, на которые разбивается запись числа. Это приводит к быстрому увеличению числа «малых» умножений. С другой стороны, только увеличение k позволяет экономить «большие» умножения. Компромисс зависит от числа разрядов умножаемых чисел.

Посмотрим на это чуть более подробно. В нашем случае x_i суть числа $0, \pm1, \ldots, \pm k$, k – число блоков, на которые разбивается запись числа, степень многочлена PQ равна $2k$, $y_i = P(x_i)Q(x_i)$ имеют примерно $2\cdot n/k$ разрядов. Поэтому формулы для восстановления коэффициентов PQ можно записать в виде

$$a_s = \left(\sum_{i=1}^{2k+1} y_i a_{is}\right)\Big/\beta,$$

где β – общее кратное чисел

$$\tau_i = (x_i-x_1)(x_i-x_2)\times\ldots \ldots\times\left(x_i-\hat{x}_i\right)\cdot\ldots\cdot(x_i-x_n).$$

Несложно убедиться в том, что все τ_i делят $(2k+1)!$ и что числа a_{is} имеют порядок $(2k+1)!$. Таким образом, в качестве β можно взять $(2k+1)!$, и количество разрядов в числах a_{is} и β примерно равно $2k\cdot\log_{10}(2k)$. (Это следует из *формулы Стирлинга*: $n!\approx\sqrt{2\pi n}(n/e)^n$ при больших n.) Итак, после осуществления $2k+1$ «большого» умножения и нахождения y_i остается найти $2k+1$ коэффициент многочлена PQ. Каждый такой коэффициент находится как сумма $2k+1$ слагаемого $y_i a_{is}$. Одно-единственное деление на $\beta = (2k+1)!$ можно произвести в самом конце. Последняя операция имеет сложность порядка $2k\cdot\log_{10}(2k)$. (Деление мы осуществляем в столбик.)

Если «малые» умножения осуществлять в столбик, то получается оценка

$$T(n) \le (2k+1)T(n/k) + \\ + (2k+1)^2 2n/k\cdot 2k\log_{10}k + 2nk\log_{10}k.$$

Можно действовать оптимальнее, разбив десятичную запись y_i на блоки по $2k\log_{10}(2k)$ разрядов и умножая блоки с помощью быстрого умножения. Можно получить такую оценку:

$$T(n) \le (2k+1)T(n/k) + \\ + (2k+1)^2 n/k^2\cdot T\left(2k\log_{10}2k\right) + 2nk\log_{10}k.$$

Конечно, процедуру вычисления коэффициентов a_{is} можно оптимизировать. Ведь промежуточные вычисления для одного коэффициента можно использовать для другого.

Хотя вопросы оптимизации алгоритмов умножения весьма интересны, их более подробное обсуждение выходит за рамки данной статьи, главная цель которой – рассказать о самом факте быстрого умножения. А разработка и оптимизация соответствующих алгоритмов, вместе с их программной реализацией, может послужить темой хорошего доклада на научной конференции учащихся (впрочем, на «взрослой» конференции тоже). Поэтому мы предлагаем читателю следующую задачу:

Задача на исследование. *Разработайте алгоритмы умножения многозначных чисел. Исследуйте вопрос о числе блоков разбиения для каждого шага.*

Впоследствии были найдены более совершенные схемы (А.Тоом, С.Кук, А.Шенхаге). А.Шенхаге и Ф.Штрассен построили эффективно работающий алгоритм умножения с числом элементарных операций $\le Cn\log n\log\log n$. Эти схемы помимо интерполяционных полиномов использовали так называемое *быстрое преобразование Фурье*. Если читатель хочет заняться предложенной задачей, то мы советуем ему обратиться к замечательной книге Д.Кнута «Искусство программирования» или написать нам по адресу kanel@mccme.ru или kanel@dnttm.ru.

Нет ничего элементарнее задачи нахождения произведения двух чисел. Этой задачей занимаются в начальной школе. Но компьютер в школе не учился! И оказалось, что для эффективного решения элементарной задачи умножения необходимы интерполяционные полиномы и довольно тонкие не вполне элементарные комбинаторные методы. Вероятно, в этом заключается одна из причин, по которым быстрое умножение было открыто сравнительно недавно.