ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сборник трудов
1963 г. Института математики СО АН СССР Выпуск 7

ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ МНОГОЧЛЕНА НА ВС

Г.А. Бекишев

Данная статья посвящена вопросу о минимальном числе шагов, за которое можно вычислить на ВС значение многочлена $f(x)$ в точке x по формуле

$$f(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n .$$

При этом предполагается, что число L машин в ВС достаточно велико. По своему содержанию в целом статья примыкает к работам [1], [2], [3], [4].

I.° Сделаем предварительно несколько уточнений относительно постановки задачи.

Пусть

$$f(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n \qquad (I)$$

– многочлен n -й степени с коэффициентами a_i, отличными от 0 и 1.

Обозначим через $\mathcal{U}=\{\mathcal{A}\}$ семейство алгоритмов вычисления $f(x)$, построенных в соответствии с формулой (I). Это значит, что каждый алгоритм семейства $\mathcal{U}$ построен с помощью операций „+" и „×" и включает в себя вычисление всех членов a_jx^j многочлена (I). В частности, в семейство $\mathcal{U}$ войдут алгоритмы вычисления $f(x)$, предусматривающие параллельное выполнение операций на каждом шаге вычислений. При этом мы

считаем, что единственным требованием, наложенным на число операций, выполняемых параллельно, является требование их выполнимости на данном шаге вычислений.

Так же, как и в работе [1], число шагов вычислений, связанное с реализацией данного алгоритма $\mathcal{A}$, будем называть его длиной.

Основная задача, которую мы себе ставим, теперь может быть сформулирована так. В семействе алгоритмов $\mathcal{U}$ найти алгоритм наименьшей длины.

ТЕОРЕМА I. Наименьшая из длин алгоритмов семейства $\mathcal{U}$ равна

$$h=h(n)=\begin{cases} n+1, & n=1,2,3. \\ \left[\log 4n\right]+\left[\log\left(n+1+2^{\left[\log n\right]}\left(\sum\limits_{\kappa=1}^{\mathcal{L}}2^{-2^{\kappa}}-1\right)\right)\right], & n\geqslant 4, \end{cases}$$

где $\mathcal{L}=\left[\frac{[\log n]}{2}\right]$, а символ $[\log m]$ обозначает целую часть двоичного логарифма числа m.

ДОКАЗАТЕЛЬСТВО. Сформулированная теорема будет доказана, если мы для данного n построим некоторый алгоритм $\mathcal{A}$ семейства $\mathcal{U}$ длины h и покажем, что всякий другой алгоритм этого семейства имеет длину, не меньшую, чем h.

В справедливости теоремы при n = I,2,3 можно убедиться непосредственной проверкой, поэтому будем считать, что $n\geqslant 4$.

Рассмотрим матрицу

$$C=\begin{pmatrix} \zeta_{10}, \zeta_{11}, \ldots, \zeta_{1n} \\ \zeta_{20}, \zeta_{21}, \ldots, \zeta_{2n} \\ \ldots\ldots\ldots\ldots \\ \zeta_{s0}, \zeta_{s1}, \ldots, \zeta_{sn} \end{pmatrix},$$

состоящую из $n+1$ столбцов, s строк и имеющую такой вид: в её j -м столбце, j = I,2,..., n, стоит двоичная запись числа nj, начиная с младших разрядов, а элементы 0 -го столбца определены равенствами:

$$\zeta_{10}=\zeta_{20}=\ldots=\zeta_{s-1,0}=1, \quad \zeta_{s0}=0 .$$

Таким образом, каждый элемент ζ_{ij} матрицы C равен либо 0, либо I, а число строк s матрицы C равно числу цифр в двоичной записи n, т.е. $s=\left[\log n\right]+1$.

С помощью матрицы C построим новую матрицу таких же размеров

$$\mathcal{D} = (\eta_{ij}) ,$$

$$(i = 1,2,\dots, s \;;\; j = 0,1,2,\dots n) ,$$

элементы которой определены равенствами:

$$\eta_{ij} = \begin{cases} \zeta_{ij}\, a_j x^{\sum_{\kappa=1}^{i} \zeta_{\kappa j} 2^{\kappa-1}} , & \text{если } j = 1,2,\dots n \\ \zeta_{ij} x^{2^i} , & \text{если } j = 0. \end{cases}$$

$$(i = 1,2,\dots s) .$$

Очевидно, что в матрице $\mathcal{D}$ ненулевых элементов будет столько же, сколько таковых в матрице C .

Строим теперь следующий алгоритм $\mathcal{A}$ вычисления $f(x)$. На первом шаге вычисляются элементы первой строки матрицы $\mathcal{D}$. В частности, будет вычислен член $a_1 x = \eta_{11}$ первого измерения в многочлене $f(x)$. На втором шаге вычисляются элементы второй строки матрицы $\mathcal{D}$. При этом будут вычислены члены $a_2 x^2 = \eta_{22}$, $a_3 x^3 = \eta_{23}$ второго и третьего измерения, соответственно. Вообще на i -м шаге вычисляются элементы i -й строки матрицы $\mathcal{D}$, причем, в качестве интересующих нас величин будут вычислены члены

$$\eta_{i,2^{i-1}} = a_{2^{i-1}} x^{2^{i-1}} , \eta_{i,2^{i-1}+1} = a_{2^{i-1}+1} x^{2^{i-1}+1} , \dots , \eta_{i,2^i} = a_{2^i} x^{2^i}$$

Наконец, на s -м шаге вычисляются элементы s -й строки матрицы $\mathcal{D}$, т.е. члены многочлена

$$\eta_{s,2^{s-1}} = a_{2^{s-1}} x^{2^{s-1}} , \dots , \eta_{sn} = a_n x^n$$

Кроме того, на втором шаге мы сложим члены a_0 и $a_1 x$, а на всяком $(i+1)$ -м шаге $(i = 2,3,\dots, s-1)$ произведем попарное сложение членов многочлена, вычисляемых на i -м шаге, и результатов сложений на всех предыдущих шагах. Последующие шаги вычислений, начиная с $(s+1)$-го, сводятся к попарным сложениям, причем, на каждом шаге выполняется столько необходимых сложений, сколько это максимально возможно.

Покажем теперь, что для длины описанного алгоритма $\mathcal{A}$ имеет место оценка, указанная в теореме.

В самом деле, нетрудно подсчитать, что после s -го шага при $n \geqslant 4$ предстоит еще сложить

$$n - (2^{[\log n]} - 1) + 2^{[\log n]-2} + 2^{[\log n]-4} + \dots + 2^{[\log n]-2\left[\frac{[\log n]}{2}\right]} + 1 =$$

$$= n + 2 + 2^{[\log n]} \left(\sum_{1 \leqslant \kappa \leqslant \left[\frac{[\log n]}{2}\right]} 2^{-2\kappa} - 1 \right)$$

слагаемых, для чего потребуется как минимум

$$h_1 = \left[\log \left(n + 1 + 2^{[\log n]} \left(\sum_{1 \leqslant \kappa \leqslant \left[\frac{[\log n]}{2}\right]} 2^{-2\kappa} - 1 \right) \right) \right] + 1$$

шагов вычислений [I] . Следовательно, общее число шагов вычислений, т.е. длина построенного алгоритма $\mathcal{A}$, при $n \geqslant 4$ выразится формулой:

$$h = h_1 + s \left[\log 4n \right] + \left[\log \left(n + 1 + 2^{[\log n]} \left(\sum_{1 \leqslant \kappa \leqslant \left[\frac{[\log n]}{2}\right]} 2^{-2\kappa} - 1 \right) \right) \right] . \qquad (2)$$

Для завершения доказательства теоремы остается показать, что полученная формула выражает наименьшую из длин алгоритмов семейства $\mathcal{A}$.

Для этой цели нам потребуется одна простая лемма.

ЛЕММА. Пусть l, m, κ – целые неотрицательные числа, причем, $l > m \geqslant \kappa \geqslant 0$. Тогда

$$\kappa + \left[\frac{l-\kappa}{2} \right] \leqslant m + \left[\frac{l-m}{2} \right] .$$

Действительно, представим число κ в виде

$$\kappa = m - j , \quad j = 0,1,2,\dots m .$$

Тогда доказательство исходного неравенства будет равносильно доказательству неравенства

$$m - j + \left[\frac{l-m+j}{2} \right] \leqslant m + \left[\frac{l-m}{2} \right]$$

для указанных выше значений j .

Последнее неравенство при $j = 0$ очевидно. Легко видеть, что оно будет верно и при замене j на $j+1$.

Следовательно, лемма доказана.

Пусть теперь $S_1, S_2, \dots, S_h$ – последовательность групп операций (групповых операций), выполняемых при реализации построенного выше алгоритма $\mathcal{A}$, соответственно, на

I,2,... h -м шаге вычислений, а $\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_q$ -аналогичная последовательность, связанная с произвольным алгоритмом $\bar{A}$ семейства $\mathcal{A}$. Надо показать, что $h \le q$.

Два алгоритма вычисления $f(x)$, принадлежащие семейству $\mathcal{A}$, будут характеризоваться, в общем случае, разным числом операций сложения и умножения. Однако, число операций сложения для любых двух алгоритмов из $\mathcal{A}$ будет одно и то же и равно n.

Пусть $|X_i|$ $(i = 1,2,\ldots,h)$ означает число операций сложения в групповой операции S_i, а $|\bar{X}_j|$ $(j = 1,2,\ldots q)$ - аналогичное число для $\bar{S}_j$.

На основании сказанного имеем

$$\sum_{i=1}^{h} |X_i| = \sum_{j=1}^{q} |\bar{X}_j| = n.$$

Покажем, что

$$\sum_{j=1}^{s} |\bar{X}_j| \le \sum_{j=1}^{s} |X_j| \quad , \qquad (3)$$

где S по-прежнему определяется равенством

$$S = [\log n] + 1 \quad .$$

Во-первых, заметим, что

$$h > S \quad \text{и} \quad q > S \quad .$$

Это следует из того, что при любом способе вычислений член j -го измерения $a_j x^j$ $(j = 1,2,\ldots n)$ многочлена $f(x)$ может быть вычислен не меньше, чем за $[\log j] + 1$ шагов [I].

Далее мы имеем

$$|\bar{X}_1| = |X_1| = 0 \quad .$$

Пусть уже $$\sum_{j=1}^{i} |\bar{X}_j| \le \sum_{j=1}^{i} |X_j|$$

для некоторого $i = 1,2,\ldots S-1$. Покажем, что тогда и

$$\sum_{j=1}^{i+1} |\bar{X}_j| \le \sum_{j=1}^{i+1} |X_j| \quad . \qquad (4)$$

В алгоритме A вычисления $f(x)$, который подробно был описан выше, за $i-1$ шагов вычисляются члены многочлена (I)

$$a_1 x, a_2 x^2, \ldots a_{2^{i-1}} x^{2^{i-1}}$$

Число их, вместе с a_0, равно 2^{i-1} и будет максимальным для всех алгоритмов из $\mathcal{A}$, ибо за $i-1$ шагов никаких других членов многочлена полностью вычислить нельзя. На i -м шаге при любом алгоритме семейства $\mathcal{A}$ может быть вычислено самое большее 2^{i-1} новых членов многочлена $f(x)$. Следовательно,

$$\sum_{j=1}^{i+1} |X_j| = \sum_{j=1}^{i} |X_j| + \left[\frac{2^{i-1} - \sum_{j=1}^{i} |X_j|}{2}\right] + 2^{i-2}$$

и

$$\sum_{j=1}^{i+1} |\bar{X}_j| \le \sum_{j=1}^{i} |\bar{X}_j| + \left[\frac{2^{i-1} - \sum_{j=1}^{i} |\bar{X}_j|}{2}\right] + 2^{i-2} .$$

Полагаем теперь

$$l = 2^{i-1}, \quad \sum_{j=1}^{i} |\bar{X}_j| = \kappa \quad , \quad \sum_{j=1}^{i} |X_j| = m \quad .$$

В силу индуктивного предположения будем иметь $\kappa \le m$. Применяя лемму, докажем неравенства (4) и (3).

Теперь можно завершить доказательство теоремы I. В самом деле, в алгоритме A после S шагов нам предстоит еще сложить

$$n+1 - \sum_{j=1}^{s} |X_j| = n+2+2^{[\log n]} \left(\sum_{1 \le \kappa \le \left[\frac{[\log n]}{2}\right]} 2^{-2\kappa} - 1 \right)$$

чисел. Следовательно, для длины алгоритма A, наряду с формулой (2), мы будем иметь равносильную формулу

$$h = S + h_1 = S + \left[\log\left(n - \sum_{j=1}^{s} |X_j|\right)\right] + 1 \quad .$$

В то же время для длины q алгоритма $\bar{A}$ имеем, очевидно,

$$q \ge S + \left[\log\left(n - \sum_{j=1}^{s} |\bar{X}_j|\right)\right] + 1 \quad .$$

Ввиду неравенства (3) ясно, что $q \ge h$.

Теорема доказана.

Представление о характере изменения определенной формулой (2) функции $h = h(n)$ легко получается из следующих замечаний. Во-первых, ясно, что функция $h(n)$ будет монотонно неубывающей. Во-вторых, для вычисления её значений в точках вида $n = 2^\rho$, $\rho = 1, 2, \ldots$, мы имеем особенно простые формулы:

$$h(2^\rho) = \begin{cases} 2\rho+1 , & \rho = 1, 2, \\ 2\rho , & \rho \geqslant 3 . \end{cases}$$

Действительно, из формулы (2) при $n = 2^\rho$, $\rho = 2, 3, \ldots$, получаем

$$h(2^\rho) = \rho + 2 + \left[\log\left(1 + \sum_{1 \leqslant \kappa \leqslant \left[\frac{\rho}{2}\right]} 2^{\rho - 2\kappa}\right)\right] = \begin{cases} \rho+3 = 2\rho+1, & \rho = 2 \\ 2\rho , & \rho \geqslant 3 . \end{cases}$$

При $\rho = 1$, в силу теоремы I, получаем

$$h(2) = 3 = 2\rho + 1 .$$

Характер роста функции $h(n)$ теперь очевиден.

2^0. Для того, чтобы судить об эффективности вычисления значений многочлена $f(x)$ на ВС по сравнению с вычислением на одной машине мы докажем следующую теорему.

ТЕОРЕМА 2. Наименьшее число операций сложения и умножения, которое потребуется для того, чтобы вычислить значение многочлена $f(x)$ в точке x по формуле (I), равно $\ell(n) = 3n - 1$.

ДОКАЗАТЕЛЬСТВО. В соответствии с формулой (I) могут быть построены различные алгоритмы вычисления $f(x)$. Любой из этих алгоритмов будет включать в себя вычисление членов многочлена $a_j x^j$ $(j = 1, 2, \ldots n)$ и их сложение. Очевидно, произвольный алгоритм вычисления $f(x)$ будет характеризоваться одним и тем же числом операций сложения, равным n. Поэтому достаточно показать, что наименьшее число операций умножения для вычисления всех членов многочлена $a_j x^j$ равно $\ell^{(y)}(n) = 2n - 1$.

При $n = 1$ это утверждение тривиально.
Пусть уже

$$\ell^{(y)}(\kappa) = 2\kappa - 1 .$$

Покажем, что

$$\ell^{(y)}(\kappa+1) = 2(\kappa+1) - 1 = 2\kappa + 1 .$$

Действительно, члены многочлена $(\kappa+1)$-й степени

$$f(x) = \sum_{i=0}^{\kappa+1} a_i x^i$$

могут быть, очевидно, вычислены с помощью $2(\kappa+1)-1 = 2\kappa+1$ операций умножения: κ операций умножения потребуется для получения степеней $x^2, x^3, \ldots, x^{\kappa+1}$ и еще $\kappa+1$ операций умножения для получения одночленов $a_j x^j$ $(j = 1, 2, \ldots, \kappa+1)$. Меньшим числом операций умножения эти вычисления проделать нельзя.

В самом деле, допустим, что $\ell^{(y)}(\kappa+1) = 2\kappa$.

При любом способе вычисления членов $a_1 x, a_2 x^2, \ldots, a_{\kappa+1} x^{\kappa+1}$ для получения $a_{\kappa+1} x^{\kappa+1}$ потребуется по меньшей мере две операции умножения: либо будет вычислена степень $x^{\kappa+1}$, а затем произведение $a_{\kappa+1} x^{\kappa+1}$, либо будет вычислено $a_{\kappa+1} x^j$ $(j < \kappa+1)$ и затем $a_{\kappa+1} x^{\kappa+1}$. Очевидно, если из совокупности операций, с помощью которых получаются члены $a_1 x, \ldots, a_{\kappa+1} x^{\kappa+1}$, удалить названные выше 2 операции, связанные с вычислением $a_{\kappa+1} x^{\kappa+1}$, то оставшиеся $2\kappa - 2$ операций умножения реализуют вычисление членов $a_1 x, a_2 x^2, \ldots, a_\kappa x^\kappa$. Последнее, однако, невозможно в силу сделанного допущения.

Теорема 2 доказана.

ЗАМЕЧАНИЕ. С помощью указанного числа ℓ операций нельзя в общем случае вычислить $f(x)$ по формуле (I) за минимальное число h шагов. Оценка минимума числа операций при этом дополнительном требовании также представляла бы интерес.

Из предыдущего можно сделать некоторые выводы. При вычислении значений многочлена $f(x)$ на одной машине наиболее выгодно, по-видимому, пользоваться схемой Горнера, т.е. формулой

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + a_n x)\ldots)) ,$$

так как при этом потребуется всего n операций сложения и n операций умножения. Однако схема Горнера в силу своего рекуррентного характера перестает быть лучшей при вычислении значений многочлена на ВС. В этом случае выгоднее строить алгоритмы вычисления $f(x)$ в соответствии с формулой (I). Величина $\frac{n}{[\log n]}$ может быть принята за меру эффективности применения ВС к вычислению значений многочленов n -й степени.

Литература

I. Бекишев Г.А. Об алгоритмах, эффективно реализуемых на вычислительных системах. Сб. "Вычислительные системы", вып. 7, I963 г., Новосибирск.

2. Бекишев Г.А. К вопросу эффективного решения простейших задач алгебры матриц на вычислительной системе, состоящей из l машин. Сб. "Вычислительные системы", I963г., Новосибирск (в печати).

3. Бекишев Г.А. О распараллеливании вычислительных алгоритмов. Сб. "Вычислительные системы", вып.5, I963, Новосибирск.

4. Бекишев Г.А. Решение одной задачи теории графов. Сб. "Вычислительные системы", вып. 6, I963 г., Новосибирск.