В Ы Ч И С Л И Т Е Л Ь Н Ы Е С И С Т Е М Ы

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

ОБ АЛГОРИТМАХ, ЭФФЕКТИВНО РЕАЛИЗУЕМЫХ НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Г.А. Бекишев

Отличительной особенностью решения задач на вычислительных системах (ВС) будет использование алгоритмов, рассчитанных на параллельное выполнение допустимого числа операций на каждом шаге вычислений [I], [3]. С точки зрения затрат времени и оборудования на решение задачи наиболее эффективны будут такие алгоритмы, которые могут быть реализованы минимальным числом машин за минимальное число шагов вычислений.

Таким образом, возникает необходимость оценки алгоритмов с указанной точки зрения. Более точная постановка вопроса дается ниже. В рамках этой постановки вопроса в статье рассматриваются задачи сложения и умножения n чисел и задача возведения числа в степень с натуральным показателем.

§ I. Постановка задачи

Алгоритмический процесс решения всякой математической задачи расчленяется на ряд достаточно элементарных шагов вычислений. Говоря об этом расчленении, мы будем понимать его с точностью до выполнения отдельных операций в случае алгоритма, предусматривающего последовательное их выполнение, и с точностью до выполнения групп операций, если алгоритм рассчитан на

параллельное выполнение операций. При этом, конечно, предполагается, что понятие операции содержательно определено. В частности, это определение операции должно игнорировать возможность её расчленения на микрооперации.

В дальнейшем в качестве набора операций можно принять набор основных операций, применяемых в современных ЭВМ. Длительность всех операций будем считать одинаковой.

Пусть $\mathfrak{A} = \{\mathcal{A}\}$ есть некоторое семейство алгоритмов, решающих данную задачу. Пусть, далее, $\mathcal{A}$ - произвольный алгоритм семейства $\mathfrak{A}$, а $S_1, S_2, \ldots, S_q$ -последовательность групп операций (групповых операций), выполняемых при реализации алгоритма $\mathcal{A}$ соответственно на I,2,..., наконец, q шаге вычислений. Число $q=q(\mathcal{A})$ назовем д л и н о й алгоритма $\mathcal{A}$. Число операций, составляющих данную групповую операцию S_i, будем называть м о щ н о с т ь ю этой групповой операции и обозначать символом $|S_i|$. Величина

$$\mathcal{N}(\mathcal{A}) = \max_{1 \leq i \leq q} |S_i|$$

будет характеризовать число машин, которое потребуется для реализации алгоритма $\mathcal{A}$.

Условимся далее называть о п т и м а л ь н ы м алгоритмом семейства $\mathfrak{A}$ такой алгоритм $\mathcal{A}$, который имеет наименьшую длину и может быть реализован наименьшим числом машин. Оптимальный алгоритм семейства $\mathfrak{A}$ характеризуется,таким образом, длиной

$$h = \min_{\mathcal{A} \in \mathfrak{A}} q(\mathcal{A})$$

и числом машин

$$\mathcal{N} = \min_{\mathcal{A} \in \mathcal{B}} \mathcal{N}(\mathcal{A}) \quad ,$$

где $\mathcal{B}$ - подсемейство всех алгоритмов семейства $\mathfrak{A}$, имеющих длину h.

Основная задача, которую мы себе ставим, теперь может быть сформулирована следующим образом. В с е м е й с т в е а л г о р и т м о в $\mathfrak{A} = \{\mathcal{A}\}$ н а й т и о п т и м а л ь н ы й а л г о р и т м и д а т ь о ц е н к у е г о п о д л и н е h и ч и с л у м а ш и н $\mathcal{N}$. Числа h и $\mathcal{N}$ иногда будут называться х а р а к т е р и с т и к а м и оптимального алгоритма.

§ 2. Сложение n чисел

Пусть $\mathfrak{A} = \{\mathcal{A}\}$ обозначает семейство всевозможных алгоритмов сложения n чисел, представленных формулой

$$x = \sum_{i=1}^{n} x_i \quad , \quad n \geqslant 2 \ .$$

Дадим оценку характеристик оптимального алгоритма сложения семейства $\mathfrak{A}$. Из этой оценки, в частности, будет следовать и самое построение оптимального алгоритма.

Начнем с доказательства одной леммы.
Пусть $\mathcal{A}$ - произвольный алгоритм рассматриваемого семейства, а

$$S_1, S_2, \ldots, S_q$$

есть последовательность групповых операций, определяемая алгоритмом $\mathcal{A}$. Имеет место следующая лемма.

ЛЕММА I. При любом $i = 1, 2, \ldots, q$ мощность групповой операции S_i удовлетворяет неравенству

$$|S_i| \leqslant 2^{q-i} \ .$$

ДОКАЗАТЕЛЬСТВО. Для S_q лемма очевидно верна, так как $|S_q| = 1$. Пусть она верна уже для групповых операций $S_q, S_{q-1}, \ldots, S_i$. Возьмем групповую операцию S_{i-1} и предположим, что ее мощность $|S_{i-1}| > 2^{q-i+1}$. В таком случае после выполнения S_{i-1} нам потребуется выполнить сложение больше 2^{q-i+1} чисел, для чего необходимо произвести более $2^{q-i+1}-1$ операций сложения. Отсюда, однако, следует, что

$$\sum_{j=i}^{q} |S_j| > 2^{q-i+1} - 1 \ ,$$

тогда как по индуктивному предположению

$$\sum_{j=i}^{q} |S_j| \leqslant 2^{q-i+1} - 1 \ .$$

Это противоречие и завершает доказательство леммы.

Применяя лемму, докажем теперь теорему о длине оптимального алгоритма семейства $\mathfrak{A}$. Символом $[a]$ будем в дальнейшем обозначать целую часть числа a, символом $\log n$ - двоичный логарифм числа n.

ТЕОРЕМА I. Оптимальный алгоритм семейства $\mathfrak{A} = \{\mathcal{A}\}$ имеет длину

$$h = h(n) = [\log(n-1)] + 1 \ . \qquad \text{(I)}$$

ДОКАЗАТЕЛЬСТВО. Разобьем множество $\mathfrak{N} = \{n\}$ натуральных чисел на группы чисел вида:

$$J_0 = \{1 = 2^0\},\ J_1 = \{2^1\},\ J_2 = \{3, 2^2\}, \ldots, J_{\rho+1} = \{2^\rho + 1, \ldots, 2^{\rho+1}\}, \ldots$$

и будем доказывать теорему индукцией по индексу групп. При $n \in J_1$ теорема верна. Предположим, что теорема верна для любого $n \in J_i$, $i \leqslant \rho$, и покажем, что тогда она будет верна и для всякого $n \in J_{\rho+1}$. В самом деле, любое $n \in J_{\rho+1}$ можно записать в виде

$$n = 2^\rho + j, \quad j = 1, 2, \ldots, 2^\rho .$$

По предположению индукции сумму 2^ρ слагаемых можно вычислить за $[\log(2^\rho - 1)] + 1 = \rho$ шагов. За такое же, или даже за меньшее число шагов, можно вычислить сумму остальных j слагаемых.

Таким образом, при указанных n сумму из n слагаемых можно вычислить за $\rho + 1 = [\log(n-1)] + 1$ шагов. В то же время эту сумму за меньшее число шагов вычислить нельзя, ибо, как это следует из леммы I, за ρ шагов можно выполнить самое большее $2^\rho - 1$ операций сложения, что меньше $n-1$ - числа операций, необходимого для сложения n чисел.

Теорема доказана.

Переходя к оценке оптимального алгоритма семейства $\mathfrak{A}$ по числу машин, мы рассмотрим предварительно задачу сложения n чисел при ограничении на число машин в ВС.

Пусть $\mathfrak{A}_L$ означает семейство тех алгоритмов сложения, представленных формулой $x = \sum_{i=1}^{n} x_i$, которые могут быть построены в расчете на ВС, состоящую из L машин. Любой из алгоритмов $\mathcal{A} \in \mathfrak{A}_L$ предусматривает, следовательно, параллельное выполнение не более L операций на каждом шаге вычислений.

Очевидно, $\mathfrak{A}_L$ будет некоторым подсемейством алгоритмов семейства $\mathfrak{A}$. Оптимальный алгоритм этого подсемейства $\mathfrak{A}_L$ будет характеризоваться некоторой длиной $h(n, L)$ и некоторым числом машин $N(n, L)$. Найдем выражение для указанных функций.

ТЕОРЕМА 2. Минимальное число шагов, за которое можно сложить n $(n \geqslant 2)$ чисел на ВС, состоящей из L машин, равно

$$h(n,L)=\begin{cases} n-1, & L=1, \\ [\log(n-1)]+1, & L\geqslant n, \\ \left[\frac{n}{L}\right]+\left[\log\left(n-L\left[\frac{n}{L}\right]+L-1\right)\right], & 2\leqslant L\leqslant n. \end{cases} \qquad (2)$$

ДОКАЗАТЕЛЬСТВО. Если $L=1$, то на каждом шаге вычислений мы можем выполнить только одну операцию сложения. Поэтому длина любого алгоритма семейства $\mathfrak{A}_L$ определяется общим числом операций, необходимым для сложения n чисел, т.е.

$$h(n,1)=n-1 .$$

При $L\geqslant n$ мы имеем $\mathfrak{A}_L\equiv\mathfrak{A}$. Применяя теорему I, получим

$$h(n,L)=[\log(n-1)]+1 , \quad L\geqslant n .$$

Остается показать, что при $2\leqslant L\leqslant n$ имеет место формула:

$$h(n,L)=\left[\frac{n}{L}\right]+\left[\log\left(n-L\left[\frac{n}{L}\right]+L-1\right)\right] \qquad (2a)$$

Пусть сначала $2\leqslant L\leqslant\left[\frac{n}{2}\right]$. Построим следующий алгоритм семейства $\mathfrak{A}_L$: сначала произведем $\left[\frac{n}{L}\right]-1$ шагов вычислений, выполняя на каждом шаге по L операций сложения. После этого нам останется сложить еще

$$n-L\left(\left[\frac{n}{L}\right]-1\right)$$

чисел. Очевидно,

$$n-L\left(\left[\frac{n}{L}\right]-1\right)<2L ,$$

то есть

$$L>\left[\frac{n-L\left(\left[\frac{n}{L}\right]-1\right)}{2}\right],$$

и потому – на основании теоремы I – указанное выше число слагаемых можно сложить самое меньшее за

$$\left[\log\left(n-L\left[\frac{n}{L}\right]+L-1\right)\right]+1$$

шагов. Общее число шагов вычислений выразится тогда формулой (2a), причем, легко видеть, что это число шагов вычислений является минимальным.

Пусть теперь $\left[\frac{n}{2}\right]<L\leqslant n$. Тогда $\mathfrak{A}_L\equiv\mathfrak{A}$, и в силу формулы (I) для указанных L имеем

$$h(n,L)=[\log(n-1)]+1 .$$

Этот же результат получается и по формуле (2a), ибо при $\left[\frac{n}{2}\right]<L\leqslant n$ имеет место равенство

$$\left[\frac{n}{L}\right]=1 .$$

Формула (2a), а вместе с ней и теорема 2, доказана.

Переходя к оценке оптимального алгоритма семейства $\mathfrak{A}_L$ по числу машин, напомним известное из теории рекурсивных функций понятие корневой, или μ - операции, с помощью которой из заданной функции от $n+1$ аргументов можно получить функцию n аргументов (корневую функцию) [2].

Пусть $g(x_1,x_2,\dots x_n,z)$ - заданная функция $n+1$ натуральных аргументов, принимающая натуральные значения. Тогда, по определению μ - операции, значение корневой функции

$$f(x_1,x_2,\dots x_n)=\mu z\{g(x_1,x_2,\dots,x_n z)=0\}$$

на n-ке $(x_1,x_2,\dots x_n)$ равно наименьшему натуральному числу z, при котором удовлетворяется уравнение, стоящее в фигурных скобках. Теперь теорема, выражающая оценку числа машин $N(n,L)$, может быть сформулирована так.

ТЕОРЕМА 3. М и н и м а л ь н о е ч и с л о м а ш и н, с п о м о щ ь ю к о т о р о г о м о ж н о с л о ж и т ь n ч и с е л н а В С, с о с т о я щ е й и з L м а ш и н, з а м и н и м а л ь н о е ч и с л о ш а г о в $h(n,L)$, р а в н о:

$$N(n,L)=\mu z\{h(n,z)=h(n,L)\} , \qquad (3)$$

г д е μ - с и м в о л, μ - о п е р а ц и и.

ДОКАЗАТЕЛЬСТВО этой теоремы очевидно. Сделаем только следующее замечание. Может показаться, что оптимальный алгоритм семейства $\mathfrak{A}_L$ будет характеризоваться L машинами, т.е. что применение μ -операции излишне. На самом деле это не так. Приведем следующий пример.

Пусть $n=28$, $L=13$. Тогда по формуле (2) имеем

$$h(28,13)=5 .$$

Применяя же формулу (3), получим

$$N(28,13)=\mu z\left\{\left[\frac{28}{z}\right]+\left[\log\left(28-z\left[\frac{28}{z}\right]+z-1\right)\right]=5\right\}=12 .$$

Легко убедиться в том, что 28 чисел можно сложить за 5 шагов с помощью 12 машин, а меньшим числом машин нельзя.

Из теоремы 3 для оценки оптимального алгоритма семейства $\mathfrak{A}$ по числу машин $\mathcal{N}$ вытекает следующая

ТЕОРЕМА 4. Оптимальный алгоритм семейства $\mathfrak{A}$ реализуется числом машин, равным

$$\mathcal{N}=\mathcal{N}(n)=\mu L\left\{\left[\frac{n}{L}\right]+\left[\log\left(n-L\left[\frac{n}{L}\right]+L-1\right)\right]=\left[\log(n-1)\right]+1\right\} . \qquad (4)$$

Найдем явное выражение для функции $\mathcal{N}(n)$. А именно, покажем, что корневая функция $\mathcal{N}(n)$, определенная формулой (4), имеет вид:

$$\mathcal{N}(n)=\begin{cases}\left[\dfrac{n-2^{[\log n]-1}-1}{2}\right]+1, & \text{если } 2^{[\log n]}<n<2^{[\log n]}+2^{[\log n]-1}; \\ n-2^{[\log(n-1)]}, & \text{если } 2^{[\log(n-1)]}+2^{[\log(n-1)]-1}\leq n\leq 2^{[\log(n-1)]+1}.\end{cases} \qquad (5)$$

I) Пусть $2^{\rho}<n<2^{\rho}+2^{\rho-1}$, $\rho=2,3,4,\ldots\ldots$.

Тогда $n=2^{\rho}+j$, $\quad j=1,2,\ldots,2^{\rho-1}-1$.

Будем различать 2 случая:

а) n - чётное, т.е. $j=2,4,\ldots,2^{\rho-1}-2$.

Возьмём $L=2^{\rho-2}+\frac{j}{2}$. Легко проверить, что

$$\left[\frac{n}{L}\right]=\left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j}{2}}\right]=3 .$$

Подставляя в формулу для $h(n,L)$ значения

$$n=2^{\rho}+j , \quad L=2^{\rho-2}+\frac{j}{2} , \quad \left[\frac{n}{L}\right]=3 ,$$

получим

$$h\left(2^{\rho}+j,2^{\rho-2}+\frac{j}{2}\right)=\left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j}{2}}\right]+\left[\log\left(2^{\rho}+j-\left(2^{\rho-2}+\frac{j}{2}\right)\left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j}{2}}\right]+2^{\rho-2}+\frac{j}{2}-1\right)\right]=\rho+1.$$

Так как при $n=2^{\rho}+j$, $j=1,2,\ldots,2^{\rho-1}-1$

$$\left[\log(n-1)\right]+1=\rho+1 ,$$

то

$$\mathcal{N}(n)\leq 2^{\rho-2}+\frac{j}{2} , \quad j=2,4,\ldots,2^{\rho-1}-2 .$$

На самом деле здесь будет иметь место знак равенства. Действительно, возьмём

$$L=2^{\rho-2}+\frac{j}{2}-1 .$$

Тогда

$$\left[\frac{n}{L}\right]=\left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j}{2}-1}\right]=\begin{cases}4 , & j=2,4 \\ 3 , & j=6,8,\ldots,2^{\rho-1}-2\end{cases}$$

и легко проверить, что при любом $j=2,4,\ldots,2^{\rho-1}-2$

$$h\left(2^{\rho}+j,2^{\rho-2}+\frac{j}{2}-1\right)=\rho+2>\rho+1 .$$

При $L<2^{\rho-2}+\frac{j}{2}-1$ равенство $h(n,L)=\rho+1$ подавно не будет выполняться и, таким образом,

$$\mathcal{N}(n)=2^{\rho-2}+\frac{j}{2} .$$

б) n - нечётное, т.е. $j=1,3,\ldots,2^{\rho-1}-1$.

Возьмем

$$L=2^{\rho-2}+\frac{j-1}{2}+1 .$$

Очевидно,

$$\left[\frac{n}{L}\right]=\left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j-1}{2}+1}\right]=\begin{cases}2 , & j=2^{\rho-1}-1, \\ 3 , & j=1,3,\ldots,2^{\rho-1}-3 .\end{cases}$$

Подставляя в выражение для функции $h(n,L)$ значения

$$n=2^{\rho}+j , \quad L=2^{\rho-2}+\frac{j-1}{2}+1 ,$$

после несложных вычислений получим

$$h\left(2^{\rho}+j,2^{\rho-2}+\frac{j-1}{2}+1\right)=\rho+1.$$

Следовательно, $$\mathcal{N}(n) \leq 2^{\rho-2} + \frac{j-1}{2} + 1 .$$

Возьмем теперь $L = 2^{\rho-2} + \frac{j-1}{2}$. Замечая, что

$$\left[\frac{n}{L}\right] = \left[\frac{2^{\rho}+j}{2^{\rho-2}+\frac{j-1}{2}}\right] = \begin{cases} 4, & j=1, \\ 3, & j=3,5,\dots,2^{\rho-1}-1, \end{cases}$$

получим

$$h\left(2^{\rho}+j,\ 2^{\rho-2}+\frac{j-1}{2}\right) = \rho+2 > \rho+1 ,$$

то есть

$$\mathcal{N}(n) = 2^{\rho-2} + \frac{j-1}{2} + 1 .$$

Таким образом, мы показали, что при $n = 2^{\rho}+j$, где $\rho = 2,3,4,\dots$; $j = 1,2,\dots,2^{\rho-1}-1$ функция $\mathcal{N}(n)$ имеет вид:

$$\mathcal{N}(n) = \begin{cases} 2^{\rho-2} + \frac{j}{2}, & j \text{ - четное;} \\ 2^{\rho-2} + \frac{j-1}{2} + 1, & j \text{ - нечетное.} \end{cases}$$

Эти две формулы можно объединить в одну. Действительно, замечая, что $\rho = [\log n]$, имеем

$$2^{\rho-2} + \frac{j}{2} = \frac{2^{\rho-1}+j}{2} = \frac{n-2^{\rho-1}}{2} = \left[\frac{n-2^{[\log n]-1}-1}{2}\right] + 1 ;$$

$$1 + 2^{\rho-2} + \frac{j-1}{2} = \frac{2^{\rho-1}+j-1}{2} + 1 = \frac{n-2^{\rho-1}-1}{2} + 1 = \left[\frac{n-2^{[\log n]-1}-1}{2}\right] + 1.$$

Следовательно,

$$\mathcal{N}(n) = \left[\frac{n-2^{[\log n]-1}-1}{2}\right] + 1 ,\quad 2^{[\log n]} < n < 2^{[\log n]} + 2^{[\log n]-1},$$

и первая часть формулы (5) доказана.

2) Пусть $$2^{[\log(n-1)]} + 2^{[\log(n-1)]-1} \leq n \leq 2^{[\log(n-1)]+1} .$$

Полагая $\rho = [\log(n-1)]$, представим n в виде:

$$n = 2^{\rho} + 2^{\rho-1} + j ,\quad \left(\begin{array}{l} \rho = 1,2,3,\dots \\ j = 0,1,2,\dots 2^{\rho-1} \end{array}\right).$$

Рассмотрим сначала случай $j = 2^{\rho-1}$, т.е. $n = 2^{\rho+1}$. Возьмем $L = 2^{\rho}$. Так как

$$h(2^{\rho+1}, 2^{\rho}) = \rho+1 ,\quad h(2^{\rho+1}, 2^{\rho}-1) = \rho+2 ,$$

то

$$\mathcal{N}(n) = \mathcal{N}(2^{\rho+1}) = 2^{\rho} = n - 2^{[\log(n-1)]},$$

то есть в этом случае формула (5) верна.

При $j = 0,1,2,\dots,2^{\rho-1}-1$ возьмем $L = 2^{\rho-1}+j$.

С учетом того, что

$$\left[\frac{n}{L}\right] = \left[\frac{2^{\rho}+2^{\rho-1}+j}{2^{\rho-1}+j}\right] = \begin{cases} 3, & j=0, \\ 2, & j=1,2,\dots,2^{\rho-1}-1, \end{cases}$$

получаем

$$h(2^{\rho}+2^{\rho-1}+j,\ 2^{\rho-1}+j) = \rho+1.$$

Отсюда следует, что $$\mathcal{N}(n) \leq 2^{\rho-1}+j .$$

С другой стороны, если взять $L = 2^{\rho-1}+j-1$, то

$$\left[\frac{n}{L}\right] = \left[\frac{2^{\rho}+2^{\rho-1}+j}{2^{\rho-1}+j-1}\right] = \begin{cases} 3, & j=0,1, \\ 2, & j=2,3,\dots,2^{\rho-1}-1 . \end{cases}$$

и мы имеем

$$h(2^{\rho}+2^{\rho-1}+j,\ 2^{\rho-1}+j-1) = \rho+2 > \rho+1 .$$

Таким образом, окончательно

$$\mathcal{N}(n) = 2^{\rho-1}+j ,\quad j = 1,2,\dots,2^{\rho-1}$$

или

$$\mathcal{N}(n) = n - 2^{[\log(n-1)]} ,\quad 2^{[\log(n-1)]} + 2^{[\log(n-1)]-1} \leq n \leq 2^{[\log(n-1)]+1} .$$

В наших рассуждениях предполагалось, что $n > 2$.

Ясно, что последняя формула сохраняет силу и при $n = 2$. Тем самым формулы (5) нами доказаны.

Заканчивая рассмотрение задачи сложения на ВС n чисел, заметим, что располагая соответствующими функциями h и $\mathcal{N}$, можно построить и сами оптимальные алгоритмы семейств $\mathfrak{A}$ и $\mathfrak{A}_L$.

§ 3. Умножение n чисел

Если иметь в виду самый общий случай умножения n чисел, то совершенно ясно, что с рассматриваемой точки зрения задача умножения n чисел на ВС ничем не будет отличаться от аналогичной задачи сложения n чисел. Поэтому все теоремы, доказанные в § 2, после замены соответствующих выражений сохраняют силу и в данном случае.

§ 4. Возведение числа в степень с натуральным показателем

Пусть $\mathfrak{A} = \{\mathcal{A}\}$ – семейство алгоритмов вычисления степени $x = x^n$ $(n \geqslant 2)$. Оценка оптимального алгоритма семейства $\mathfrak{A}$ вытекает из теоремы 5.

ТЕОРЕМА 5. Оптимальный алгоритм семейства $\mathfrak{A}$ имеет длину

$$h = h(n) = [\log(n-1)] + 1$$

и реализуется числом машин, равным

$$\mathcal{N} = \begin{cases} 1, & \text{если } r = 1, 2, \\ 2, & \text{если } r > 2, \end{cases}$$

где r – число единиц в двоичной записи числа n.

ДОКАЗАТЕЛЬСТВО. Оценка длины оптимального алгоритма вычисления степени x^n следует из теоремы I, сформулированной применительно к случаю умножения n чисел. Поэтому нам остается доказать лишь справедливость оценки для $\mathcal{N}$.

Примем следующий план доказательства: сначала покажем, что при любом n для вычисления степени x^n за минимальное число шагов h двух машин будет достаточно. Затем покажем, что с помощью одной машины за h шагов степень x^n может быть вычислена тогда и только тогда, когда двоичная запись показателя n содержит не более двух единиц.

Действительно, пусть

$$n = \xi_{p+1}\xi_p \ldots \xi_2\xi_1, \quad \xi_i = \begin{cases} 1 \\ 0 \end{cases}$$

есть двоичная запись числа n. Очевидно, $[\log n] = p$. Построим следующий алгоритм вычисления x^n: на первом шаге вычислим x^2, на втором – x^4 и $\xi_2 \cdot x^{\sum_{k=1}^{2}\xi_k 2^{k-1}}$, на i-м $(i \leqslant p)$ – x^{2^i} и $\xi_i x^{\sum_{k=1}^{i}\xi_k 2^{k-1}}$. Наконец, на $(p+1)$-м шаге вычисляем степень $\xi_{p+1} x^{\sum_{k=1}^{p+1}\xi_k 2^{k-1}}$.

Само собою разумеется, что при $n = 2^p$ $(p+1)$-й шаг отпадает и, таким образом, при любом n построенный алгоритм вычисляет степень x^n за минимальное число шагов $h = [\log(n-1)] + 1$, причем число операций, параллельно выполняемых на каждом шаге вычислений, не превышает двух. В то же время, если n таково, что $r = 1$ или $r = 2$, то степень x^n за h шагов может быть вычислена одной машиной. Покажем теперь, что верно и обратное: если степень x^n за h шагов вычисляется одной машиной, то число r единиц в двоичной записи n не превышает 2. Пусть

$$x^{i_1}, x^{i_2}, \ldots, x^{i_h}, \quad i_1 < i_2 < \ldots < i_h = n$$

есть последовательность степеней, полученная в результате вычисления x^n. Разобьем множество $\mathfrak{N} = \{n\}$ натуральных чисел на группы

$$J_0 = \{1\},\ J_1 = \{2\},\ J_3 = \{3, 2^2\}, \ldots\ J_h = \{2^{h-1}+1, \ldots 2^h\}, \ldots$$

Тогда необходимо $i_k \in J_k$ $(k = 1, 2, \ldots, h)$.

Доказываемое положение, очевидно, верно при $n \in J_1$. Допустим, что оно имеет место также для всякого $n \in J_i$, $i \leqslant h-1$. Возьмем некоторое число $i_h = n \in J_h$. Если $x^n = x^{i_h}$ вычисляется за h шагов одной машиной, то степень $x^{i_{h-1}}$ одной машиной вычисляется за $h-1$ шагов, и потому, в силу предположения индукции, двоичное разложение i_{h-1} содержит либо одну, либо две единицы.

Если для i_{h-1} число r равно 1, то $i_{h-1} = 2^{h-1}$. В этом случае необходимо $i_1 = 2$, $i_2 = 4$, $i_3 = 8, \ldots i_{h-2} = 2^{h-2}$, и потому $n = i_h = 2^{h-1} + 2^j$, $0 \leqslant j \leqslant h-1$ т.е. двоичное разложение числа n содержит либо одну, либо две единицы.

Если в двоичной записи числа i_{h-1} содержится две единицы, т.е. $i_{h-1} = 2^{h-2} + 2^j$, $j = 0, 1, 2, \ldots, h-3$, то могут представиться 2 случая:

а) двоичная запись i_{h-2} содержит одну единицу: $i_{h-2} = 2^{h-2}$;

б) двоичная запись i_{h-2} содержит две единицы: $i_{h-2} = 2^{h-3} + 2^j$, $j = 0, 1, 2, \ldots, h-4$.

В первом случае степень $x^{i_h} = x^n$ равна

$$x^{2^{h-2}+2^j} \cdot x^{2^{h-2}} = x^{2^{h-1}+2^j} \quad \text{или} \quad x^{2^{h-2}+2^j} \cdot x^{2^{h-2}+2^j} = x^{2^{h-1}+2^{j+1}}$$

$$(j = 0, 1, 2, \ldots, h-3),$$

т.е. двоичная запись $i_h = n$ содержит две единицы.

Во втором случае степень $x^{i_n} = x^n$ также может быть получена не иначе, как по формулам:

$$x^{i_h} = x^{i_{h-1}} \cdot x^{i_{h-1}} \quad \text{или} \quad x^{i_h} = x^{i_{h-1}} \cdot x^{i_{h-2}}.$$

Первая из этих формул дает

$$x^{i_h} = x^{2^{h-1}+2^{j+1}}, \qquad j = 1, 2, \ldots, h-3.$$

Вторая из этих формул, очевидно, возможна лишь при $i_{h-1} = 2^{h-2} + 2^{h-3}$, и в этом случае

$$x^{i_h} = x^{2^{h-2}+2^{h-3}} \cdot x^{2^{h-3}+2^j} = x^{2^{h-1}+2^j}, \qquad j = 0, 1, 2, \ldots, h-4.$$

Таким образом, во всех случаях двоичная запись числа n содержит не более двух единиц.

Теорема доказана.

В процессе доказательства было указано также построение оптимального алгоритма вычисления x^n. Заметим, что построенный алгоритм не является, вообще говоря, единственным. Например, степень x^{22} может быть вычислена двумя минимальными "цепочками":

$$x^2;\ x^4;\ x^6,\ x^8;\ x^{16};\ x^{22},$$

$$x^2;\ x^3, x^4;\ x^8;\ x^{11};\ x^{22}.$$

Здесь степени x^i, вычисляемые на разных шагах, разделены точкой с запятой, на одном шаге - запятой.

В заключение отметим одно следствие доказанной теоремы.

СЛЕДСТВИЕ. Минимальное число шагов, за которое может быть вычислено значение одночлена ax^n в точке x $(a \neq 0, 1)$, равно $h = [\log n] + 1$.

Минимальное число машин, которое для этого потребуется, равно:

$$N = \begin{cases} 1, & \text{если} \quad r = 1, 2 \\ 2, & \text{если} \quad r > 2, \end{cases}$$

где r - число единиц в двоичной записи числа n.

Полученные выше оценки дают возможность судить об эффективности применения ВС для решения рассмотренного класса задач.

Литература

I. Евреинов Э.В., Косарев Ю.Г. О возможности построения вычислительных систем высокой производительности. Изд-во СО АН СССР, Новосибирск, I962 г.

2. Успенский В.А. Лекции о вычислимых функциях. М., I960 г.

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