

自然数 K 进制下运算速度的初步研究

队员：叶天扬 魏博 王明苑 指导老师：凌惠明
南京市金陵中学

摘要：本文研究自然数在不同进制下进行加法和乘法运算的速度，通过构建函数模型定量比较了不同进制下进行加法和乘法运算的时间的多少。最终发现二进制是进行加法的最优进制，二进制和三进制在进行乘法时都要优于其他进制，而它们之间又各自有一定的适应范围。

The Primary Study of the Calculating Speed of Natural Number in Different Scale Systems

Abstract: This article studied the speed of addition and multiplication of natural number in different scale systems, the times of addition and multiplication in different scale systems have been compared quantificationally through a function model presented. The conclusion is that binary system is the best for addition, binary system and trinary system are better than other scale systems for multiplication, and they each has a suitable range.

引言

现代社会信息繁杂，每天有如潮水般的数据需要计算，这都要依赖计算机。在数字计算机的发展中，二进制的应用功不可没。计算机中不管多复杂的运算，最终都归结到数字电路中二进制的加减法。数字电路的两种稳定状态代表了二进制的零和一。但随着科技的发展，人类或许会应用存在三种甚至三种以上稳定状态的器件来制作计算机，那时计算机的基础就不一定是二进制了。如果选择的进制恰当，就能够缩短计算时间。纵观各种进制可以发现，大进制方便记数，进行的运算次数少，可运算复杂，符号繁多；小进制运算快捷，但记数困难，运算的次数多。本文研究的目标就是构造一个函数来衡量不同进制下运算的速度，权衡利弊，找出对某一范围内的数进行运算最适合的进制。

考虑到问题的复杂性，本文中只讨论自然数的加法和乘法。

本文中的 k 为大于等于 2 的正整数。

由于本文讨论的是在大范围的计算中进制的性质，因此直接用事件概率进行计算。

首先，要对运算进行一个分析。“数”是自然界本身的一种抽象存在，人类把它用一个符号表示了出来，就是我们平时打交道的数，在一定进制下的运算就是按一定规则（根据这种进制的加法表和乘法表）进行符号转换，转换出来的符号所对应的数就是要运算的结果。

下面来考察最基本的运算，两个一位数的加法，乘法，称之为单位运算。

根据上面的分析，运算的过程就是符号转换，可以认为，单位运算就是一个查表的过程。在加法（乘法）表中找出两个加数（乘数）所在的行与列，在它们行列交会处的那个数字就是这次运算的结果。平等起见，假设各进制在运算时“查表”的速度是一样的，则不同进制下进行运算所需的时间正比于这个进制下加法（乘法）表的大小。又，既然认为加法与乘法实质上都是查表的符号转换，查什么样的表对时间并没有影响，可以认为在某一进制下进行一位数的加法和乘法运算的时间是相同的。易知， k 进制下的加法（乘法）表的大小是 $k \times k$ （包含零）。假设在“一进制”下查 1×1 的加法表和乘法表的时间为单位时间 t ，则 k 进制下的单位运算时间为

$$t_k = k^2 t \quad (1)$$

又知，若 A 在 k 进制下表示为 m 位数，则

$$k^{m-1} \leq A \leq k^m - 1 \quad m = [\log_k A] + 1 \quad (2)$$

1 加法

k 进制下，一个 m 位数和一个 n 位数相加，不妨设 $n \leq m$ 。

这里研究加法的运算量采用竖式，但不采用平常竖式加法的顺序。

先不考虑进位，只将上下两数相加，再把进位的数加上去。

例如：计算 $678+125$ 。

$$\begin{array}{r} 6 \ 7 \ 8 \\ + \ 1 \ 2 \ 5 \\ \hline \end{array}$$

竖式加法的顺序是 $8+5=13$ 进 1

$$2+7=9 \quad 9+1=10 \text{ 进 } 1$$

$$6+1=7 \quad 7+1=8$$

此处加法的顺序为 $8+5=13$ $7+2=9$ $6+1=7$

$$\text{进 } 1 \quad 9+1=10$$

$$\text{进 } 1 \quad 1+7=8$$

因为加法进位只能进 1，若两数相加进位再加一不可能再进位 ($((k-1)+(k-1)+1 < 2k)$)，所以在运算总是两两进行的情况下运算总数不变，不影响运算时间。

n 位数与 m 位数相加首先要进行 $l_1 = n$ 次单位运算，然后考虑进位运算的次数。

先考虑第一次进位，两个一位数 a 、 b 相加，由于对称性， a 、 b 出现 $0 \sim (k-1)$ 这 k 个数的概率是相同的。若 a 、 b 相加不进位，则 $a+b$ 必为 $0 \sim (k-1)$ 之一。不定方程 $a+b=p \quad 0 \leq p \leq k-1$ 的非负整数解共有 $\sum_{p=0}^{k-1} C_{p+2-1}^{2-1} = \sum_{p=0}^{k-1} C_{p+1}^1 = C_{k+1}^2$ 组。考虑 (a, b) 的排列，共有 k^2 种情况。

所以不进位的概率为 $p(\text{不进位}) = \frac{C_{k+1}^2}{k^2} = \frac{k+1}{2k}$.

进位的概率为 $p(\text{第一次进位}) = 1 - \frac{k+1}{2k} = \frac{k-1}{2k}$

所以 n 次单位运算中进位的个数应为 $n(\frac{k-1}{2k})$ 。

下面考虑第二次进位。第二次进位与第一次不同，因为进位上来的只能为 1，所以它只能为 $C+1$ 的形式。又由对称性， C 为 $0 \sim (k-1)$ 的概率相等

(以十进制为例。末位得到 1 的加法序列有

(0, 1) (1, 0) (2, 9) (9, 2) (3, 8) (8, 3) (4, 7) (7, 4) (5, 6) (6, 5)

末尾得到 2 的加法序列有

(0, 2) (2, 0) (1, 1) (3, 9) (9, 3) (4, 8) (8, 4) (5, 7) (7, 5) (6, 6)

个数相同)

此时发生进位当且仅当 $C=k-1$ 。

所以第二次进位的概率为 $p(\text{第二次进位}) = \frac{1}{k}$ 。

第二次进位的位数为 $n(\frac{k-1}{2k})\frac{1}{k}$ 。

同理可得，以后发生进位的概率也为 $\frac{1}{k}$ 。

进位的总数为 $\sum_{i=0}^{\infty} n(\frac{k-1}{2k})(\frac{1}{k})^i = n(\frac{k-1}{2k})(\frac{1}{1-\frac{1}{k}}) = n(\frac{k-1}{2k})(\frac{k}{k-1}) = \frac{n}{2}$

所以进位导致的运算要进行 $l_2 = \frac{n}{2}$ 次

m 位数加上 n 位数进行的运算总数为

$$l_1 + l_2 = n + \frac{n}{2} = \frac{3n}{2} \quad (3)$$

这是个有趣的结论，竟然只与数位有关，与 k 无关。

对于两个数 A, B ， $A \leq B$ ，由 (1) (2) (3) 知，在 k 进制下计算其加法共需运算次数 $\frac{3([\log_k A]+1)}{2}$ 次，共需时间

$$\frac{3([\log_k A]+1)}{2} k^2 t \quad (4)$$

对一个确定范围的 A ，寻找 k 使耗时最小。这个函数对 k 相当复杂，对 A 却比较简单，这里从另一面考虑，对一个确定的 k ，去寻找它所适合的 A 。

若对某一范围内的 A ， k 进制比 $(k+1)$ 进制耗时少，则

$$\frac{3([\log_k A]+1)}{2} k^2 t \leq \frac{3([\log_{k+1} A]+1)}{2} (k+1)^2 t$$

$$\Leftrightarrow ([\log_k A]+1)k^2 \leq ([\log_{k+1} A]+1)(k+1)^2$$

此处进行一个近似，用连续函数代替离散函数

$$\text{令 } [\log_k A] \approx \log_k A = \frac{\ln A}{\ln k} \quad [\log_{k+1} A] \approx \log_{k+1} A = \frac{\ln A}{\ln(k+1)}$$

$$\text{则} \quad ([\log_k A] + 1)k^2 \leq ([\log_{k+1} A] + 1)(k+1)^2$$

$$\Leftrightarrow \ln A \left(\frac{k^2}{\ln k} - \frac{(k+1)^2}{\ln(k+1)} \right) \leq 2k+1$$

$$\text{令 } f(x) = \frac{x^2}{\ln x} \quad \text{则} \quad f'(x) = \frac{2x \ln x - \frac{1}{x} \times x^2}{(\ln x)^2} = \frac{x(2 \ln x - 1)}{(\ln x)^2} > 0 \text{ 对 } x \geq 2 \text{ 成立}$$

所以在 $x \geq 2$ 时， $f(x)$ 严格单调增。

$$\text{即} \left(\frac{k^2}{\ln k} - \frac{(k+1)^2}{\ln(k+1)} \right) < 0$$

所以等价的不等式左边为负数，右边为正数，显然成立。

令 $t(k)$ 为 k 进制下加法运算的时间。则由上面的分析可知

$$\text{对 } \forall A \in N^* \quad t(2) \leq t(3) \leq \dots \leq t(k)$$

综上，二进制是进行加法最好的进制。

2 乘法

从最简单的情况入手，考虑一位数与 m 位数相乘。

先要进行单位运算 m 次。单位运算结果可能为一位数，可能为两位数，设在 k 进制下单位运算的结果为两位数的概率为 p_k ，这个 p_k 可通过乘法表求得，表 1 用计算机求出了前一些 p_k 。

表 1 k 与 p_k 关系

k	p_k	k	p_k	k	p_k	k	p_k
2	0	7	0.448979592	12	0.638888889	17	0.712802768
3	0.111111111	8	0.515625	13	0.644970414	18	0.731481482
4	0.25	9	0.543209877	14	0.673469388	19	0.736842105
5	0.32	10	0.58	15	0.688888889	20	0.7525
6	0.416666667	11	0.603305785	16	0.703125	21	0.757369615

对于 k 较大的情况，可以根据表 2 所示 $k \times k$ 的乘法表，考察乘积不进位的情况，求出 p_k 的近似值。易知第 0 列 ($0 \times A$) k 个数全部不进位，第 1 列 ($1 \times A$) k 个数全部不进位，第 2 列有 $\left[\frac{k-1}{2} \right] + 1$ 个数不进位，延续上面的近似，用连续函数代替离散函数 $\left[\frac{k-1}{2} \right] + 1 \approx \frac{k}{2}$ ，依次类推。则不进位的概率

表2 k进制下的乘法表

*	0	1	2	...	k-2	k-1
0	0	0	0		0	0
1	0	1	2		k-2	k-1
2	0	2				
...						
k-2	0	k-2				
k-1	0	k-1				

$$\begin{aligned} \bar{p}_k &= 1 - p_k = \frac{1}{k^2} \left(k + k + \left[\frac{k-1}{2} \right] + 1 + \left[\frac{k-1}{3} \right] + 1 + \dots + \left[\frac{k-1}{k-2} \right] + 1 + \left[\frac{k-1}{k-1} \right] + 1 \right) \\ &\approx \frac{1}{k^2} \left(k + k + \frac{k}{2} + \frac{k}{3} + \dots + \frac{k}{k-2} + \frac{k}{k-1} \right) \\ &= \frac{1}{k} \left(2 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k-2} + \frac{1}{k-1} \right) \end{aligned}$$

$$\begin{aligned} \text{又知 } \int_1^{k-1} \frac{1}{x} dx &> \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k-2} + \frac{1}{k-1} > \int_1^k \frac{1}{x} dx - 1 \\ \Leftrightarrow \ln(k-1) &> \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k-2} + \frac{1}{k-1} > \ln k - 1 \end{aligned}$$

由此就可以求出 p_k 的范围，对于比较大的 k ，

$$\text{此处取 } p_k = 1 - \frac{\ln k + \ln(k-1) + 3}{2k} \text{ 即此范围的中间数}$$

$\begin{array}{r} * \quad 6 \ 7 \ 8 \ 9 \ m \text{位数} \\ \hline \quad \quad \quad 9 \ 1 \text{位数} \\ \quad \quad \quad 8 \ 1 \\ + \quad \quad \quad 7 \ 2 \\ \hline \quad \quad \quad 8 \ 0 \ 1 \\ + \quad \quad \quad 6 \ 3 \\ \hline \quad \quad \quad 7 \ 1 \ 0 \ 1 \\ + \quad \quad \quad 5 \ 4 \\ \hline \quad \quad \quad 6 \ 1 \ 1 \ 0 \ 1 \end{array}$	$\begin{array}{r} * \quad 6 \ 7 \ 8 \ 9 \ m \text{位数} \\ \hline \quad \quad \quad 1 \ 1 \text{位数} \\ \quad \quad \quad 9 \\ + \quad \quad \quad 8 \\ \hline \quad \quad \quad 8 \ 9 \\ + \quad \quad \quad 7 \\ \hline \quad \quad \quad 7 \ 8 \ 9 \\ + \quad \quad \quad 6 \\ \hline \quad \quad \quad 6 \ 7 \ 8 \ 9 \end{array}$
--	--

图1 一位数乘m位数的结果加起来的过

下面考察一位数相乘的结果加起来的过。这里抛弃原有竖式的加法顺序，采用积逐次相加的顺序（见图1），不影响计算量。可以看到在加法中上一行的末位是直接移下来的，所以对于一位数和两位数，计算的有效数位为零位或一位。这期间要进行 m 次加法，有 $m-1$ 个加数，由上面的分析知这其中有 $p_k(m-1)$ 个两位数， $(1-p_k)(m-1)$ 个一位数。由（3）知要进行加法运算 $\frac{3}{2} \times 1 \times p_k(m-1) + \frac{3}{2} \times 0 \times (1-p_k)(m-1) = \frac{3}{2} p_k(m-1)$ 次。

综上，一位数乘 m 位数需进行乘法运算 m 次，加法运算 $\frac{3}{2} p_k (m-1)$ 次。

以下分析在 k 进制下一个 m 位数和一个 n 位数相乘的情况。

从理论上分析，求出来的函数应该是 $m n$ 的对称式。

首先要进行 mn 次一位数乘法运算，这相当于进

行 n 个一位数乘 m 位数的计算，需进行乘法 $l_1 = mn$

次，加法 $l_2 = n \times \frac{3}{2} p_k (m-1)$ 次运算。

再看下面的加式相加，共有 n 个数，要进行 $n-1$ 次加法，有 $n-1$ 个加数（见图 2）。一位数乘 m 位数可能结果为 m 位数或 $m+1$ 位数，在末位直接下移的加法中，

有效数位为 $m-1$ 位或 m 位，此处采用了简化处理，认为一位数乘 m 位数结果为 $m+1$ 位数的概率仍为 p_k （实际上应该比 p_k 大一些，因为首位数字不可能为零，但这个近似还是在允许范围内的，尤其是后来发现这样近似的结果是一个对称式，符合理论上的要求）。

所以有 $p_k (n-1)$ 个 $m+1$ 位数 $(1-p_k) (n-1)$ 个 m 位数。由（3）知加式中共需进行运算次数

$$\begin{aligned} l_3 &= \frac{3m}{2} p_k (n-1) + \frac{3(m-1)}{2} (1-p_k)(n-1) \\ &= \frac{3(n-1)}{2} [p_k m + (1-p_k)(m-1)] = \frac{3}{2} (n-1)(m-1+p_k) \end{aligned}$$

综上， m 位数乘 n 位数，共需进行运算

$$\begin{aligned} l_1 + l_2 + l_3 &= mn + \frac{3n}{2} p_k (m-1) + \frac{3}{2} (n-1)(m-1+p_k) \\ &= mn(1 + \frac{3}{2} p_k) + \frac{3}{2} (m-1)(n-1) - \frac{3}{2} p_k \end{aligned} \quad (5)$$

正好是 m, n 的对称式。

所以，由（2）（5）知，对两个数 A, B ，在 k 进制下计算其乘法需要的运算次数为

$$([\log_k A] + 1)([\log_k B] + 1)(1 + \frac{3}{2} p_k) + \frac{3}{2} ([\log_k A])([\log_k B]) - \frac{3}{2} p_k$$

由（1）知，所需的运算时间为

$$\left[([\log_k A] + 1)([\log_k B] + 1)(1 + \frac{3}{2} p_k) + \frac{3}{2} ([\log_k A])([\log_k B]) - \frac{3}{2} p_k \right] k^2 t \quad (6)$$

又因为在实际情况中， $A B$ 多属于同一个范围，简单起见，不妨令 $A=B$ 。

再用连续函数取代离散函数， $[\log_k A] \approx \log_k A = \frac{\ln A}{\ln k}$

则所需的计算时间为

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \quad (\text{m位}) \\ * \quad \quad \quad 1 \ 1 \ 1 \quad (\text{n位}) \\ \hline 1 \ 1 \ 1 \ 1 \ 1 \quad n \\ 1 \ 1 \ 1 \ 1 \ 1 \quad \text{个} \\ 1 \ 1 \ 1 \ 1 \ 1 \quad \text{数} \\ \hline \end{array}$$

图 2 加式相加

$$\left[\left(\frac{\ln A}{\ln k} + 1 \right)^2 \left(1 + \frac{3}{2} p_k \right) + \frac{3}{2} \left(\frac{\ln A}{\ln k} \right)^2 - \frac{3}{2} p_k \right] k^2 t \quad (7)$$

利用上面的思想，寻找对某个确定的 k ，它所适合的 A 。

由 (7) 知，若 k 比 $k+1$ 更适合某一范围的 A ，则对这一范围的 A 有

$$\begin{aligned} & \left[\left(\frac{\ln A}{\ln k} + 1 \right)^2 \left(1 + \frac{3}{2} p_k \right) + \frac{3}{2} \left(\frac{\ln A}{\ln k} \right)^2 - \frac{3}{2} p_k \right] k^2 t \leq \\ & \left[\left(\frac{\ln A}{\ln(k+1)} + 1 \right)^2 \left(1 + \frac{3}{2} p_{k+1} \right) + \frac{3}{2} \left(\frac{\ln A}{\ln(k+1)} \right)^2 - \frac{3}{2} p_{k+1} \right] (k+1)^2 t \\ \Leftrightarrow & \left[\left(\frac{\ln A}{\ln k} \right)^2 \left(\frac{5}{2} + \frac{3}{2} p_k \right) + 2 \left(\frac{\ln A}{\ln k} \right) \left(1 + \frac{3}{2} p_k \right) + 1 \right] k^2 \leq \\ & \left[\left(\frac{\ln A}{\ln(k+1)} \right)^2 \left(\frac{5}{2} + \frac{3}{2} p_{k+1} \right) + 2 \left(\frac{\ln A}{\ln(k+1)} \right) \left(1 + \frac{3}{2} p_{k+1} \right) + 1 \right] (k+1)^2 \\ \Leftrightarrow & \left[\frac{\left(\frac{5}{2} + \frac{3}{2} p_k \right) k^2}{(\ln k)^2} - \frac{\left(\frac{5}{2} + \frac{3}{2} p_{k+1} \right) (k+1)^2}{(\ln(k+1))^2} \right] (\ln A)^2 + 2 \left[\left(\frac{1 + \frac{3}{2} p_k}{\ln k} \right) k^2 - \left(\frac{1 + \frac{3}{2} p_{k+1}}{\ln(k+1)} \right) (k+1)^2 \right] \ln A \leq 2k + 1 \end{aligned}$$

以二进制和三进制的比较为例。

取 $k=2$ ，由于 k 比较小，所以 p_k 使用实际数值。

$$p_2 = 0, p_3 = \frac{1}{9}$$

$$\left[\frac{\left(\frac{5}{2} + \frac{3}{2} p_2 \right) 2^2}{(\ln 2)^2} - \frac{\left(\frac{5}{2} + \frac{3}{2} p_3 \right) 3^2}{(\ln 3)^2} \right] (\ln A)^2 + 2 \left[\left(\frac{1 + \frac{3}{2} p_2}{\ln 2} \right) 2^2 - \left(\frac{1 + \frac{3}{2} p_3}{\ln 3} \right) 3^2 \right] \ln A \leq 2 \times 2 + 1$$

$$\Leftrightarrow (20.8136 - 19.8848)(\ln A)^2 + 2(5.7707 - 9.5575) \ln A \leq 5$$

$$\Leftrightarrow -0.70113 \leq \ln A \leq 8.8548$$

$$\Leftrightarrow 0.49602 \leq A \leq 7008.112$$

(由于选取的有效数位较少，与后文计算机算出来的结果略有出入，本文以计算机算出来的结果为准)

由此，求出了二进制乘法优于三进制乘法的范围

$$0.5411988944 \leq A \leq 6423.1009727330 \quad (8)$$

下面利用完全相同的方法，编程算出了其他相邻进制比较下所适合的范围。(见表 3)

由这张表可以看到，除 2 进制与 3 进制的比较外，其余比较求出来的范围都没有实际意义。这里用另一种方法进行比较，选取几个不同范围内有代表性的 A ，对于不同的 k 利用公式 (7) 计算其时间。用程序计算后，列成表格如下 (见表 4)。因为单位时间不影响比较，此处我们认为单位时间 $t=1$ 。

表3 相邻进制比较表 (p_k 取用实际值)

2 compare with 3	0.5411988944	$\leq A \leq$	6423.1009727330
3 compare with 4	0.0911847947	$\leq A \leq$	0.4859988020
4 compare with 5	0.1308296590	$\leq A \leq$	0.3997087792
5 compare with 6	0.0981193423	$\leq A \leq$	0.4703136977
6 compare with 7	0.1171215877	$\leq A \leq$	0.3655654868
7 compare with 8	0.0754583260	$\leq A \leq$	0.4555899733
8 compare with 9	0.0834543192	$\leq A \leq$	0.3847067303
9 compare with 10	0.0656224679	$\leq A \leq$	0.4157047754
10 compare with 11	0.0646738915	$\leq A \leq$	0.3876011628
11 compare with 12	0.0493320983	$\leq A \leq$	0.4276332168
12 compare with 13	0.0633592585	$\leq A \leq$	0.3409315275

表4 不同A, k时间表 (p_k 取用实际值)

A \ k	2	3	4	5	6	7	8	9	10
10	141	158	216	283	372	461	572	684	811
100	499	519	670	847	1080	1308	1594	1877	2196
1000	1077	1090	1377	1715	2159	2591	3130	3659	4255
10000	1876	1872	2339	2888	3610	4309	5179	6031	6988
100000	2896	2865	3554	4367	5433	6463	7743	8993	10395
1000000	4136	4068	5023	6150	7627	9052	10820	12545	14476
10000000	5597	5483	6746	8238	10192	12076	14410	16686	19231
100000000	7279	7108	8723	10631	13129	15536	18515	21417	24660
1000000000	9182	8945	10953	13329	16438	19432	23133	26738	30763
10000000000	11305	10992	13437	16333	20119	23762	28265	32649	37540

由表4可以看出二进制和三进制在我们所考察的范围内, 计算时间都远远小于其他较大的进制。而由(8)知, 二进制和三进制又各有一定的适应范围。

考虑到二进制在计算时间上与三进制相差无几, 我们认为二进制与三进制都是计算乘法最优的进制。

结束语

本文着力讨论了不同进制下进行加法和乘法运算的速度, 并定量比较了不同进制下进行运算所需要的时间。最终结论是, 二进制是进行加法运算最优的进制, 二进制和三进制在进行乘法运算时都要由于其他进制, 而三进制在乘数大于6424时优于二进制。再考虑到二进制和三进制是符号系统最为简单的两种进制, 最优运算进制可谓是当之无愧了。可见计算机选用二进制作为运算的基础, 不仅考虑到了数字电路的稳定性, 在进制运算上也是非常有优势的。