用对数表计算乘法

把乘法转化为加法的计算方法

Namishu

这份讲义从竖式乘法和乘法表出发，说明对数表如何把乘法问题转化为加法问题。读者会看到对数表、指数表、底数选择和计算精度之间的关系。

目录

1 用竖式计算乘法

乘法可以利用竖式进行计算。例如 512×648，过程如下所示。

$$
\begin{array}{ccccccc}
 & & & & 5 & 1 & 2 \\
 & & & \times & 6 & 4 & 8 \\
\hline
 & & & 4 & 0 & 9 & 6 \\
+ & & 2 & 0 & 4 & 8 & \\
+ & 3 & 0 & 7 & 2 & & \\
\hline
 & 3 & 3 & 1 & 7 & 7 & 6
\end{array}
$$

我们可以数一数，一共做了几次乘法，几次加法？

如果我们的任务是，做很多这样的乘法题。我们知道怎么计算，但是竖式的计算过程很麻烦，每列一次竖式，其中都包括很多次一位数的加法和乘法。

我们没有计算器，也不想每次都算得那么累。更重要的是，我们的目标并不是“算对一道题”，而是高效地计算大量乘法题。

2 乘法表行不行

学乘法的时候，我们背诵过九九乘法表，像下面这样。

$$
\begin{array}{lllllllll}
1 \times 1 = 1 \\
2 \times 1 = 2 & 2 \times 2 = 4 \\
3 \times 1 = 3 & 3 \times 2 = 6 & 3 \times 3 = 9 \\
4 \times 1 = 4 & 4 \times 2 = 8 & 4 \times 3 = 12 & 4 \times 4 = 16 \\
5 \times 1 = 5 & 5 \times 2 = 10 & 5 \times 3 = 15 & 5 \times 4 = 20 & 5 \times 5 = 25 \\
6 \times 1 = 6 & 6 \times 2 = 12 & 6 \times 3 = 18 & 6 \times 4 = 24 & 6 \times 5 = 30 & 6 \times 6 = 36 \\
7 \times 1 = 7 & 7 \times 2 = 14 & 7 \times 3 = 21 & 7 \times 4 = 28 & 7 \times 5 = 35 & 7 \times 6 = 42 & 7 \times 7 = 49 \\
8 \times 1 = 8 & 8 \times 2 = 16 & 8 \times 3 = 24 & 8 \times 4 = 32 & 8 \times 5 = 40 & 8 \times 6 = 48 & 8 \times 7 = 56 & 8 \times 8 = 64 \\
9 \times 1 = 9 & 9 \times 2 = 18 & 9 \times 3 = 27 & 9 \times 4 = 36 & 9 \times 5 = 45 & 9 \times 6 = 54 & 9 \times 7 = 63 & 9 \times 8 = 72 & 9 \times 9 = 81
\end{array}
$$

我们可以做一个更大的乘法表，比如“千千”乘法表（1000 以内整数相乘），然后写在一张纸上。如果纸张不够写，我们就一页一页地写，这可能会写成一本很厚的书。如果以后遇到 1000 以内的整数乘法，就从这本书里找答案。

这个想法非常好。因为这样的“千千乘法表”只需要做一次，后面就可以重复利用这个结果。这一次虽然麻烦，但是以后就省事了。

接下来，照着九九乘法表的样子，我们试着做一个千千乘法表。

第 1 行是 1×1

第 2 行是 2×1，2×2

第 3 行是 3×1，3×2，3×3

以此类推……

第 1000 行是 1000×1，1000×2，1000×1000

我们算一下，这个“千千乘法表”包含多少个算式。只要把每行的算式数数量加起来，即

$$1 + 2 + 3 + \cdots + 1000 = ?$$

注意 $1, 2, \ldots, 1000$ 是一个公差为 1 的等差数列，可以用等差数列的求和公式计算它们的总和。

$$1 + 2 + 3 + \cdots + 1000 = \frac{1000 \times (1 + 1000)}{2} = 500500$$

接下来我们估计一下，需要多少页纸。如果一页纸写 100 个式子，那么这本书总的页数是：

$$500500 \div 100 = 5005$$

一般来说，一本 500 页的书已经比较厚了。如果页数再多，就不方便装订在一起，应该分开装订，那么我们要装订大约 $5005 \div 500 = 10.01 \approx 10$ 本书。

也就是说，我们需要 10 本书（每本 500 页）来记录这个千千乘法表。然后每次从这 10 本书中找到我们想要的答案。

等等，想象一下要从 10 本书中查答案，简直是太麻烦了。虽然不用计算乘法，但是这样很浪费时间！还不如用竖式呢。

3 把乘法变成加法

回过来想一想，乘法为什么麻烦？因为要列竖式。竖式为什么麻烦？因为竖式里面有更多的乘法和加法。还有进位，也可以看成是加法。

如果不做乘法，只做加法，那就轻松多了。那么，能不能把一个乘法问题，转换为一个加法问题呢。注意，不能直接使用加法。例如计算 512×648 的时候，把 512 加 648 次，这种方式肯定不行。

我们要找一种“绕个弯，但更省力”的办法。打个比方，把“乘法”想象成一颗大树上的果子，我们伸手去摘这个果子，发现摘不到，跳起来都摘不到。这个时候怎么办呢？如果有一个梯子就好了。我们爬上梯子，再伸手摘果子。

接下来我们要找的“梯子”，就是对数。利用这个梯子，把乘法问题简化为加法问题。

先观察一个简单的例子。

$$16 \times 32 = ?$$

先看第一个乘数 16，它是 4 个 2 相乘，即

$$16 = 2 \times 2 \times 2 \times 2$$

为了简化式子，我们把它记作

$$16 = 2^4$$

其中 4 写在 2 的右上角，叫作“指数”。

再看第二个乘数 32。我们发现，32 是 5 个 2 相乘，即

$$32 = 2 \times 2 \times 2 \times 2 \times 2 = 2^5$$

再回到我们的问题：16×32 一共有几个 2 相乘？

先是 5 个 2 相乘，再接着 4 个 2 相乘，因此答案是 $4 + 5 = 9$ 个 2 相乘。用公式来表示，就是下面这样。

$$16 \times 32 = 2^4 \times 2^5 = 2^{4+5} = 2^9$$

如果我们知道 2^9 等于几，那么这个乘法问题就变成了一个“简单的”加法问题。在上面的式子中，我们只做一次加法，即 $4 + 5 = 9$。

接下来的问题是，我们怎么知道 2^9 等于几？

可以用“乘法表”的思路。那就是，提前把这些结果算出来，然后写在纸上。这样在以后的计算中，我们只要翻答案就行了。

比如像下面这样。

n	1	2	3	4	5	6	7	8	9	10	11	12	...
2^n	2	4	8	16	32	64	128	256	512	1024	2048	4096	...

现在我们找一找 2^9 等于多少。

上面的表格有两行。我们先看第一行。从 1 开始往右边数，找到数字 9 对应的列，即

n	...	**9**	...
2^n	...	512	...

再看第二行，512 就是 2^9。于是，我们得到最后的答案：

$$16 \times 32 = 2^4 \times 2^5 = 2^9 = 512$$

如果不放心，我们可以用竖式来验算一下。

$$\begin{array}{rrrr} & & 1 & 6 \\ & \times & 3 & 2 \\ \hline & & 3 & 2 \\ + & 4 & 8 & 0 \\ \hline & 5 & 1 & 2 \end{array}$$

我们发现，用这两种计算方法得到的结果是一样的。

4 做一个练习

按照上面的思路，我们来算一下 128×32。

第一步，看第一个乘数 128，它是几个 2 相乘？注意，我们不用自己算，直接从上面的表中找答案。这次从第二行开始，找到 128 对应的列。

n	...	**7**	...
2^n	...	128	...

再看第一行，它告诉我们，$128 = 2^7$，也就是 7 个 2 相乘。为了方便描述，我们把 7 称作 128 的“对数”（以 2 为底），用下面的符号表示。

$$\log_2 128 = 7$$

其中 log 符号表示要计算的是“对数”。log 右下角的数字 2 表示底数。这个式子意味着 $2^7 = 128$，其中的 2 就是底数。

第二步，看第二个乘数 32，它是几个 2 相乘？换句话说，32 的对数（以 2 为底）是几？

继续看上面的表格第二行，找到 32 所在的列。

n	...	**5**	...
2^n	...	32	...

我们得到

$$\log_2 32 = 5$$

第三步，计算 $7+5=12$。

第四步，通过查表计算 2^{12}。在上表中看第一行，找到 12 所在的列。

n	...	**12**	...
2^n	...	4096	...

再看第二行，得到 $2^{12}=4096$。即，$128\times 32=4096$。

最后，我们想一想为什么是这样。

128 是 7 个 2 相乘，32 是 5 个 2 相乘。那么，128×32 一共是 $7+5=12$ 个 2 相乘。

12 个 2 相乘用符号表示就是 2^{12}。通过查表，我们知道 $2^{12}=4096$。因此，$128\times 32=4096$。

在整个计算过程中，我们只做了一次加法，其他的计算都是查表得到。这个表我们已经提前算好了。用这样的方式计算，比用竖式计算更省事。

对数表计算乘法

用对数表计算乘法时，先查出两个乘数的对数，再把两个对数相加，最后反查指数表得到乘积。核心变化是：

$$\text{乘法} \longrightarrow \text{加法}.$$

5 注意计算的误差

考虑 32×45。先看第一个乘数，查表得到 32 的对数是 5。

n	...	**5**	...
2^n	...	32	...

但是，第二个乘数 45 的对数就查不到了。如下表所示，在第二行中 32 的下一个数是 64，而 45 在 32 和 64 之间。

n	...	5	6	...
2^n	...	32	64	...

说明这个表还不够完整。我们需要提前把 45 的对数计算出来（也就是 $\log_2 45$），并且填在这个表中，像下面这样。

n	...	5	**?**	6	...
2^n	...	32	45	64	...

观察上面的表，我们猜它的结果应该是在 5 和 6 之间，因此它是个小数。我们就不要手算 $\log_2 45$ 了，还是用计算器吧。

$$\log_2 45 \approx 5.49$$

结果保留两位小数，四舍五入。

n	...	5	**5.49**	6	...
2^n	...	32	45	64	...

接下来把 32 的对数和 45 的对数相加。

$$\log_2 32 + \log_2 45 = 5 + 5.49 = 10.49$$

最后，我们要计算 $2^{10.49}$。继续查表，看表的第一行。我们想找到 10.49。

n	1	2	3	4	5	5.49	6	7	8	9	10	11	12	...
2^n	2	4	8	16	32	45	64	128	256	512	1024	2048	4096	...

但是没有这个数字。说明这个表格还是不完整。$2^{10.49}$ 手算也不好算，还是用计算器吧。

$$2^{10.49} \approx 1438$$

结果保留整数，四舍五入。那么 $32 \times 45 \approx 1438$。

用竖式验算一下。

$$\begin{array}{rrrrr} & & & 4 & 5 \\ & & \times & 3 & 2 \\ \hline & & & 9 & 0 \\ + & 1 & 3 & 5 & 0 \\ \hline & 1 & 4 & 4 & 0 \end{array}$$

我们发现不对！

想一想原因是什么。是计算出现错误了吗？还是说这个计算方法错了？

这个方法在数学原理上是完全正确的。误差并不是来自“对数这个概念”，而是来自我们在实际操作中只能使用有限精度的对数值和指数值。

注意 $\log_2 45 \approx 5.49$，这是一个近似的结果。如果算得更精确一点，比如保留四位小数，可以得到 $\log_2 45 \approx 5.4919$。再把 32 和 45 的对数相加 $\log_2 32 + \log_2 45 = 5 + 5.4919 = 10.4919$，最后计算 $2^{10.4919} \approx 1440$。因此，计算的精度越高，得到的结果就越准确。

在实际应用中，误差是不可避免的。如果误差太大，就提高计算精度。所以，我们暂时先不管误差。接下来，我们思考一下如何把这个“表格”做得更完整，便于我们在实际中去使用它。

6 做一个对数表

先简化问题。考虑这样的问题，用对数计算 100 以内的两个整数相乘，比如 35×36，76×99，1×100 等等。虽然有些题目一眼能看出答案，我们仍然要用“对数”的方法计算。这样做的目的是为了检验我们方法对所有的情况都适用。

第一步是计算乘数的对数。我们先做一张「对数表」，如下图所示。

n	1	2	3	4	5	6	7	8	$\cdots$	100
$\log_2 n$	?	?	?	?	?	?	?	?	...	?

第一行的数字从 $1-100$ 代表乘数，第二行就是这个乘数的对数（以 2 为底）。我们知道，

$$2^0=1,\quad 2^1=2,\quad 2^2=4,\quad 2^3=8,\quad 2^4=16,\quad 2^5=32,\quad 2^6=64$$

那么

$$\begin{aligned}
2^0&=1 &&\Rightarrow \log_2 1=0\\
2^1&=2 &&\Rightarrow \log_2 2=1\\
2^3&=8 &&\Rightarrow \log_2 8=3\\
2^4&=16 &&\Rightarrow \log_2 16=4\\
2^5&=32 &&\Rightarrow \log_2 32=5\\
2^6&=64 &&\Rightarrow \log_2 64=6
\end{aligned}$$

把这些结果先填到表格中去。

n	**1**	**2**	3	**4**	5	6	7	**8**	$\cdots$	100
$\log_2 n$	**0**	**1**	?	**2**	?	?	?	**3**	...	?

其他的对数，比如 $\log_2 3$ 的结果应该是 1 到 2 之间的某个小数；$\log_2 5, \log_2 6, \log_2 7$ 应该是 2 到 3 之间的小数。手动计算这些数非常麻烦，我们还是用计算器吧。

计算结果就不在这里展示了。总之，这个对数表有两行，第一行是 n 的值，从 $1,2$ 一直到 100；第二行是 $\log_2 n$ 的值，从 $\log_2 1, \log_2 2$ 一直到 $\log_2 100$。

有了这个对数表，就可以查询 $1-100$ 之间任意整数的对数值。

7 做一个指数表

接下来我们还需要一个指数表，用来查询最后的答案。比如我们要计算 77×36。

第一步，查对数表。

77 的对数是 $\log_2 77\approx6.27$

36 的对数是 $\log_2 36\approx5.17$

第二步，对数相加。

$$\log_2 77 + \log_2 36 \approx 6.27 + 5.17 = 11.44$$

第三步，通过查表来计算 $2^{11.44}$。这个表我们把它叫做「指数表」，表的第一行是 x 值，例如 11.44，表的第二行是 2^x。

问题来了，x 可以是小数。这个指数表中，应该列出哪些 x 的数值？如果把所有小数都列出来，说不定比“千千乘法表”更复杂。

我们想一下，现在只考虑 100 以内的乘法。那么最大的对数是 $\log_2 100$，用计算器算一下，我们发现 $\log_2 100 \approx 6.64$。

我们计算的是两个数相乘，那么两个数都不会超过 100。也就是说，这两个乘数的“对数”相加不会超过 $6.64 + 6.64 = 13.28$。

如果我们以 0.01 作为基本单位。x 的值从 0 开始，每次增加 0.01，即

$$0.00, 0.01, 0.02, \ldots, 13.26, 13.27, 13.28$$

然后计算 2^x 的值，即

$$2^{0.00}, 2^{0.01}, 2^{0.02}, \ldots, 2^{13.26}, 2^{13.27}, 2^{13.28}$$

这样一来，就得到了指数表。

x	0.00	0.01	0.02	...	13.26	13.27	13.28
2^x	$2^{0.00}$	$2^{0.01}$	$2^{0.02}$	...	$2^{13.26}$	$2^{13.27}$	$2^{13.28}$

第二行具体的数值，暂时先不计算。但是不用担心，可以用计算器。

在计算之前，我们先数一下第一行中 x 一共有多少个数值。这样方便我们评估大致的工作量。

x 的范围从 0.00 开始，每次增加 0.01，一直到 13.28。如果把所有的数字乘以 100，只会改变这些数的大小，而不会改变它们的数量。所以它的数量应该等于 $0, 1, 2, \ldots, 1328$ 的数量，那么就是 1329 个数值。

总结一下，为了利用对数计算百以内乘法。我们需要两张表：对数表和指数表。其中对数表用来查两个乘数的对数，它一共有有 100 项；指数表用来查询 2^x，其中 x 是两个对数的和，它一共有 1329 项。两个表加起来一共就是 1429 项。

这么看起来工作量是可以接受的。但问题是，这样的工作量只能解决百以内的乘法。如果我们的目标是计算 1000 以内两个数相乘，那么这两张表会不会非常大？有没有可能像“千千乘法表”一样，需要十本书才能写得下？

那就算一下。假设要计算 1000 以内两个数相乘。对数表只需要计算 $1 - 1000$ 的对数，这里总共有 1000 项。

注意到，最大的乘数是 1000。我们用计算器得到它的对数是 $\log_2 1000 \approx 9.97$；那么两个乘数的对数相加不会超过 $9.97 + 9.97 = 19.94$。这说明指数表的范围是 $0.00, 0.01, \ldots, 19.94$，一共需要 1995 项。

因此，两个表加在一起是 2995 项。

回顾“千千乘法表”，我们已经算过它有 500500 项。这么一对比，对数表和指数表的项数只有“千千乘法表”的百分之六。

$$2995 \div 500500 \approx 0.006 = 6\%$$

这个方法是可行的！

接下来我们就可以把这两张表算出来。但是别着急。我们还可以再优化一下，让我们的工作量更少。

8 换一个底数

请注意，前面的对数表和指数表相关，都是以 2 为底数。我们想一想，一定要以 2 为底数才可以吗？以 10 为底数行不行？我们来探索一下。

问题还是跟之前一样。我们要计算 100 以内两位数的乘法。按照前面的思路，我们需要填两张表。

第一张是对数表。它的作用是查乘数的对数。要注意，这里的底数是 10。乘数的范围是 $1-100$，所以这个表有 100 项，如下所示。

n	1	2	3	4	5	6	7	8	9	10	⋯	100
$\log_{10} n$	?	?	?	?	?	?	?	?	?	?	...	?

我们知道

$$10^0=1,\quad 10^1=10,\quad 10^2=100$$

那么

$$\begin{aligned} 10^0&=1 &&\Rightarrow \log_{10} 1=0\\ 10^1&=10 &&\Rightarrow \log_{10} 10=1\\ 10^2&=100 &&\Rightarrow \log_{10} 100=2 \end{aligned}$$

把这几个结果先填上去，我们得到

n	**1**	2	3	4	5	6	7	8	9	**10**	⋯	**100**
$\log_{10} n$	**0**	?	?	?	?	?	?	?	?	**1**	...	**2**

观察发现，$\log_{10} 2, \log_{10} 3, \ldots, \log_{10} 9$ 的结果在 $0-1$ 之间；$\log_{10} 11, \log_{10} 12, \ldots, \log_{10} 99$ 的结果在 $1-2$ 之间。具体的值就要用到计算器了。不过暂时先不用计算，我们知道这个表长什么样子就行了。

第二张是指数表。它的作用是计算“次方”，即 10^x，其中 x 的数值来自两个“对数的和”。100 以内最大的乘数是 100，它的对数是 2。那么 x 的最大值不会超过 $2+2=4$。

如果以 0.01 为单位，那么 x 的取值就是 $0.00, 0.01, 0.02, \ldots, 3.99, 4.00$。指数表就是下面的样子。

x	0.00	0.01	0.02	...	3.99	4.00
10^x	1	?	?	...	?	10000

等一等，我们真的需要计算这么多 x 值吗？例如上面最后一列 $10^{4.00}=10^4=10000$，这是显然的。

接下来看 $10^{3.99}$。可以把指数的整数部分进行分离，把 3.99 写成 $3+0.99$，然后利用公式 $10^{a+b}=10^a\times 10^b$，我们得到

$$10^{3.99}=10^3\times 10^{0.99}=1000\times 10^{0.99}$$

于是只要查 $10^{0.99}$ 的值就行了。也就是说，3.99 这一列不需要。

同样的道理，3.98 这一列也不需要。依次类推，所有 x 值大于或等于 1 的列都不需要。另外，$10^0=1$ 也是显然的。

因此，指数表包含下面这些列就可以了。我们数一下，一共是 99 项。

x	0.01	0.02	...	0.98	0.99
10^x	?	?	...	?	?

做个总结。利用对数计算 100 以内的乘法。我们考虑两种方案。

第一种方案，以 2 为底数制作对数表和指数表。前面我们已经算过了，对数表有 100 项，指数表有 1329 项，一共是 1429 项。

第二种方案，以为 10 为底数制作对数表和指数表。对数表有 100 项，指数表有 99 项，总共是 199 项。

第二种方案的项数显然小于第一种方案的项数。这说明以 10 为底数的方案更好。到这里，我们已经发现了一个重要规律：底数选得好，表就会小很多。

9 完整的方案

我们已经知道 100 以内的乘法如何用对数和查表的方式来计算。但是，我们不满足于此。我们希望挑战一下用对数计算 1000 以内的乘法。

我们需要两张表。

第一张表是对数表，它记录了 $1-1000$ 的对数（以 10 为底数）。

n	1	2	3	4	5	6	7	8	9	10	⋯	1000
$\log_{10} n$	?	?	?	?	?	?	?	?	?	?	...	?

注意，这个表格有 1000 列，A4 纸一行写不下，还要想一想怎么排版。

我们打算把表格竖着排版。

n	$\log_{10} n$
1	
2	
...	
50	

n	$\log_{10} n$
51	
52	
...	
100	

n	$\log_{10} n$
101	
102	
...	
150	

一个表有 50 项，一页并排 5 个表格。那么一页就能写 250 项。一共有 1000 项，那么 4 页纸就可以写完对数表。

第二个表是指数表。它计算 10^x。我们以 0.001 为基本单位，x 的数值从 0.001 开始，然后是 $0.002, 0.003, \dots, 0.999$，一共 999 项。

另外，我们发现 x 的值在 $0.001-0.999$ 之间，它们都是 0 开头的数字。为了节约排版的空间，我们把 0 省略，只写小数部分，即 $001, 002, \dots, 999$，实际上代表 $0.001, 0.002, \dots, 0.999$。

x	10^x
001	
002	
...	
050	

x	10^x
051	
052	
...	
100	

x	10^x
101	
102	
...	
150	

表格的也是竖着排列。一个表格 50 项，一页 5 个表格，总共 4 页纸就够了。然后就是用计算器把这些结果算出来。这些工作比较枯燥，我们没必要手动算。用编程的方式，我们已经算好了，接下来可以直接使用。

接下来我们验证一下。

计算 512×648。

- **第一步** 查对数表。

$$\log_{10} 512 \approx 2.7093, \quad \log_{10} 648 \approx 2.8116$$

- **第二步** 把两个对数相加。

$$2.7093 + 2.8116 = 5.5209$$

- **第三步** 查指数表。

$$10^{0.5209} \approx 10^{0.521} = 3.3189$$

- **第四步** 计算答案。

$$\begin{aligned} 10^{5.5209} &\approx 10^5 \times 10^{0.521} \\ &\approx 100000 \times 3.3189 \\ &= 331890 \end{aligned}$$

我们已经用竖式算过 512×648，它的准确结果是 331776。用对数算的近似结果是 331890。它跟正确结果差了 $331890 - 331776 = 114$。这个差异还是比较明显的。这么看来，当前的精度并不太实用。但在提高精度后，它在历史上曾经被广泛使用。

例如，计算对数的时候，试试保留 6 位小数。

利用计算器，我们得到

$$\log_{10} 512 \approx 2.709270, \quad \log_{10} 648 \approx 2.811575$$

那么 $2.709270 + 2.811575 = 5.520845$。

再计算 $10^{0.520845} \approx 3.317760$，那么最后的结果是

$$\begin{aligned} 10^{5.520845} &= 10^5 \times 10^{0.520845} \\ &\approx 100000 \times 3.317760 \\ &= 331776.0 \end{aligned}$$

这样就得到了更准确的答案。

精度的作用

对数表方法的数学原理是精确的，误差主要来自查表时保留的小数位数。表格越精细，最后得到的乘积就越接近准确值。

10 对数表的历史小故事

在对数出现之前，人们已经会做乘法，也会列竖式。但在实际应用中，尤其是天文学、航海和工程计算中，人们要面对的是大量多位数的乘法和除法。这些计算非常耗时，而且极容易出错。

17 世纪初，苏格兰数学家约翰・纳皮尔（John Napier）正是被这些繁琐的计算问题所困扰。他反复思考一个问题：能不能把“困难的运算”变成“更容易的运算”？在当时，人们普遍认为加法和减法比乘法和除法容易得多，于是纳皮尔产生了一个大胆的想法——有没有办法把乘法转化为加法？

纳皮尔注意到一个重要的事实：如果两个数写成同一底数的幂，那么它们相乘时，指数只需要相加。例如，

$$a^m \times a^n = a^{m+n}.$$

这意味着，只要为每一个数找到一个合适的“指数表示”，复杂的乘法就可以转化为简单的加法。

基于这一思想，纳皮尔在 1614 年发表了世界上第一部系统介绍对数的著作：《奇妙的对数定律描述》（Mirifici Logarithmorum Canonis Descriptio）。

这是人类历史上第一张系统的对数表。他给每一个数配上一个新的数值，这个数值后来被称为“对数”。通过查表，人们可以把乘法和除法问题转化为对数的加减法，再通过反查表得到结果。

不过，纳皮尔最初使用的对数形式，与我们今天熟悉的形式并不完全相同。虽然思想是正确的，但在实际使用中仍然不够直观。

这时，英国数学家亨利・布里格斯（Henry Briggs）登场了。布里格斯敏锐地意识到，如果把对数的底数选为 10，那么计算会更加方便，因为人们日常使用的正是十进制系统。

1615 年，布里格斯专程拜访纳皮尔，两人进行了深入的交流，并很快达成共识：纳皮尔提出了对数的思想，而以 10 为底的对数形式，更适合大规模实际计算。此后，布里格斯花费多年时间，亲自计算并整理出了高精度的常用对数表。

从此，对数真正成为了一种实用工具。在计算器出现之前的三百多年里，天文学家、航海家和工程师们，几乎都依赖对数表来完成复杂计算。

回顾我们前面的讨论可以发现，对数的发明并不是为了制造抽象概念，而是一个非常现实的问题的答案：如何把大量繁琐的乘法，变成尽可能少的加法。这正是纳皮尔和布里格斯留给我们的思想遗产。