The Multiplication Table Problem Revisited

Richard P. Brent
Australian National University and
CARMA, University of Newcastle

2 Oct 2019

Short abstract

The *multiplication table problem* of Erdős concerns the asymptotics of the function $M(n)$ that counts the number of distinct products in an $n \times n$ multiplication table. We describe algorithms for evaluating $M(n)$ in quadratic time, and mention a new algorithm that takes subquadratic time. These algorithms have been used to compute $M(n)$ for various $n \leqslant 2^{30}$.

We also describe two Monte Carlo algorithms for estimating $M(n)$. These algorithms are practical for much larger n. In comparing the efficiencies of the Monte Carlo algorithms, we were led to consider a function $T(n)$, defined to be the number of products that occur exactly twice in the $n \times n$ table. We consider the numerical evidence that $T(n) \asymp M(n)$.

The talk describes joint work with Carl Pomerance (Dartmouth), David Purdum and Jonathan Webster (Butler). For more, see our preprint arXiv:1908.04251.

Outline

- History
- Two algorithms for exact computation - naive and incremental
- A subquadratic exact algorithm
- Two approximate (Monte Carlo) algorithms - Bernoulli and product trials
- Comparing Monte Carlo Algorithms – the function $T(n)$
- Numerical results
- Avoiding factoring - algorithms of Bach and Kalai

I only have 20 minutes, so have to refer you to our preprint arXiv:1908.04251 for details.

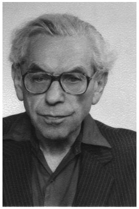
The multiplication table for $n = 7$

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

We sometimes write rows in the reverse order. A set of distinct entries is shown in blue.

The function $M(n)$ is the number of distinct entries in the $n \times n$ multiplication table: $M(7) = 25$ in the example shown.

Some contributors


Erdős

Linnik

Vinogradov

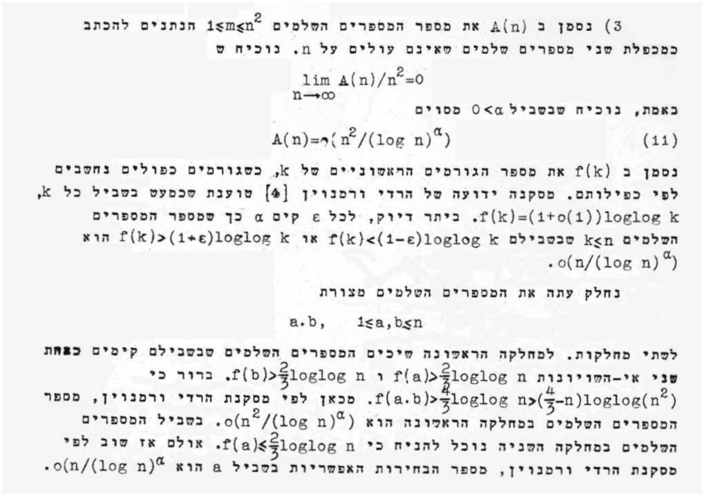
Tenenbaum

Ford

None of these well-known mathematicians was able to give an asymptotic formula for $M(n)$, though Ford [*Annals,* 2008] found the correct order of magnitude.

Erdős's first paper on $M(n)$

Here is an excerpt from Erdős's 1955 paper. The text runs right to left, but the mathematics runs left to right!



3) נסמן ב $A(n)$ את מספר המספרים השלמים $1\leq m\leq n^2$ הנתנים להכתב
כמכפלת שני מספרים שלמים שאינם עולים על n. נוכיח ש

$$\lim_{n\to\infty} A(n)/n^2=0$$

באמת, נוכיח שבשביל $0<\alpha$ מסוים

$$A(n)=o(n^2/(\log n)^{\alpha}) \qquad (11)$$

נסמן ב $f(k)$ את מספר הגורמים הראשוניים של k, כשגורמים כפולים נחשבים
לפי כפילותם. מסקנה ידועה של הרדי ורמנוין [4] טוענת שכמעט בשביל כל k,
$f(k)=(1+o(1))\log\log k$. ביתר דיוק, לכל ε קים α כך שמספר המספרים
השלמים $k\leq n$ שבשבילם $f(k)<(1-\varepsilon)\log\log k$ או $f(k)>(1+\varepsilon)\log\log k$ הוא
$o(n/(\log n)^{\alpha})$.

נחלק עתה את המספרים השלמים מצורת

$$a.b, \quad 1\leq a,b\leq n$$

לשתי מחלקות. למחלקה הראשונה שיכים המספרים השלמים שבשבילם קימים כאחת
שני אי-השויונות $f(a)>\frac{2}{3}\log\log n$ ו $f(b)>\frac{2}{3}\log\log n$. ברור כי
$f(a.b)>\frac{4}{3}\log\log n>(\frac{4}{3}-n)\log\log(n^2)$. מכאן לפי מסקנת הרדי ורמנוין, מספר
המספרים השלמים במחלקה הראשונה הוא $o(n^2/(\log n)^{\alpha})$. בשביל המספרים
השלמים במחלקה השניה נוכל להניח כי $f(a)\leq\frac{2}{3}\log\log n$. אולם אז שוב לפי
מסקנת הרדי ורמנוין, מספר הבחירות האפשריות בשביל a הוא $o(n/(\log n)^{\alpha}$.

Some notation

$f \ll g$ means that $f = O(g)$.

$f \gg g$ means that $g = O(f)$.

$f \asymp g$ means that $f \ll g$ and $f \gg g$.

$f \sim g$ means that $\lim_{n\to\infty} f(n)/g(n) = 1$.

$\log(x)$ is a logarithm to any base.

$\ln(x)$ is the natural logarithm.

$\mathbb{E}(x)$ is the expectation of x.

$\mathbb{P}\mathrm{rob}(x)$ is the probability of x.

History

There is an easy lower bound

$$M(n) \geqslant \sum_{\text{prime } p \leqslant n} p \gg \frac{n^2}{\ln n}\,.$$

Ford (2008) found the exact order-of-magnitude

$$M(n) \asymp \frac{n^2}{(\ln n)^c (\ln\ln n)^{3/2}}\,, \tag{1}$$

where c is a mysterious constant that appeared in earlier work of Erdős (1960):

$$c = \int_1^{1/\ln 2} \ln t \, \mathrm{d}t = 1 - \frac{1 + \ln\ln 2}{\ln 2} \approx 0.0861.$$

Asymptotic behaviour unknown

It is not known if there exists

$$K = \lim_{n\to\infty} \frac{M(n)(\ln n)^c(\ln\ln n)^{3/2}}{n^2}\,.$$

Ford's result only shows that the $\liminf$ and $\limsup$ are positive and finite.

Thus, there is some interest in computing exact or approximate values of $M(n)$ for "large" values of n.

Exact computation of $M(n)$ – the naive algorithm

It is easy to write a program to compute $M(n)$ for small values of n. We need an array A of size n^2 bits, indexed 1 to n^2, which is initially cleared. Then, using two nested loops, set

$$A[i \times j] \leftarrow 1 \;\text{ for }\; 1 \leqslant i \leqslant j \leqslant n.$$

Finally, count the number of bits in the array that are 1 (or sum the elements of the array). The time and space requirements are both of order n^2.

The inner loop of the above program is essentially the same as the inner loop of a program that is sieving for primes. Thus, the same tricks can be used to speed it up. For example, multiplications can be avoided as the inner loop sets bits indexed by an arithmetic progression. The sieving can easily be parallelised.

Exact computation - the incremental algorithm

The naive algorithm takes time $\asymp n^2$ to compute one value $M(n)$. If we want to tabulate $M(k)$ for $1 \leqslant k \leqslant n$, the time required is $\asymp n^3$. A more efficient approach is to compute the differences $D(n) := M(n) - M(n-1)$.

Assuming we know $M(n-1)$, we need to consider the products $m \times n$, for $1 \leqslant m \leqslant n$. Let $\delta(n)$ denote the number of these products that have already occurred in the table. The number of elements that have *not* already appeared is $D(n) = n - \delta(n)$. Thus, it is sufficient to compute $\delta(n)$ in order to compute $D(n)$ and then $M(n)$.

Computing $\delta(n)$

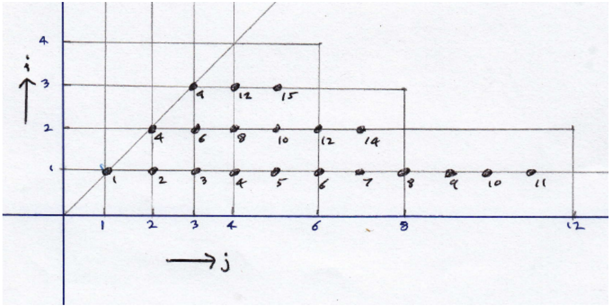
Assume we know the divisors of n (these can be computed by trial division up to $\sqrt{n}$). Suppose that $g|n$, and let $h = n/g$. By symmetry we can assume that $g \leqslant \sqrt{n} \leqslant h$.

Can we express $m \times n$ as a product that already occurred in the table? If $m = i \times j$ and $n = g \times h$, then $m \times n = ij \times gh = ih \times jg$. If $ih < n$ and $jg < n$, then the product $ih \times jg$ has already occurred. Observe that $ih < n$ iff $i < g$ and $jg < n$ iff $j < h$.

Thus, to compute $\delta(n)$, we need to count the unique products ij with $0 < i < g$ and $0 < j < n/g$, for each divisor $g \leqslant \sqrt{n}$ of n. This can be done by sieving in an array of size n (the naive algorithm required size n^2).

If this is implemented so that work is not duplicated where rectangles of size $g_1 \times n/g_1$ and $g_2 \times n/g_2$ overlap, then the work is bounded by the number of lattice points under the hyperbola $xy = n$. Thus, we can compute $\delta(n)$ in (worst case) time $O(n \log n)$ and space $O(n)$.

Example: $n = 24$



If $n = 24$, the relevant divisors are $g \in \{2, 3, 4\}$.
If $g = 2$, we sieve over $(i, j) \in \{1\} \times \{1, 2, \ldots, 11\}$.
If $g = 3$, we sieve over $(i, j) \in \{1^*, 2\} \times \{1^*, 2, 3, 4, 5, 6, 7\}$.
If $g = 4$, we sieve over $(i, j) \in \{1^*, 2^*, 3\} \times \{1^*, 2^*, 3, 4, 5\}$.
Starred entries give duplicates so may be omitted.
We conclude that $m \times n$ is already in the table iff
$m \in \{1, 2, \ldots, 11, 12, 14, 15\}$, so $\delta(n) = 14$.
e.g. $12 \times n = 16 \times 18$, $14 \times n = 16 \times 21$, $15 \times n = 20 \times 18$.

Tabulating $M(1), \ldots, M(n)$

Using the "incremental" algorithm to compute the differences $D(n) = M(n) - M(n-1)$, we can compute *all* of $M(1), \ldots, M(n)$ in time $O(n^2 \log n)$ and space $O(n)$ (not counting space for the output).

This is much better than time $O(n^3)$ and space $O(n^2)$ using the naive algorithm!

Another advantage of the incremental algorithm is that the sieve has size n, smaller than that for the naive algorithm, and more likely to fit in cache. (In both cases the sieve size can be reduced by segmentation, but the incremental algorithm still wins.)

OEIS A027417

The OEIS sequence A027417 is defined by $a_n = M(2^n - 1) + 1$. Until recently, only $a_0, \ldots, a_{25}$ were listed.

Using parallel implementations of the naive and incremental algorithms, we have extended the computation to a_{30}.

n	a_n	$4^n/a_n$
1	2	2.0000
2	7	2.2857
3	26	2.4615
4	90	2.8444
...	...	...
26	830751566970327	5.4211
27	3288580294256953	5.4779
28	13023772682665849	5.5328
29	51598848881797344	5.5860
30	204505763483830093	5.6376

A subquadratic algorithm

Using the ideas of the incremental algorithm, and some optimisations that depend on whether or not n is B-smooth with $B = L(n)^{1/\sqrt{2}}$, where $L(n) := \exp(\sqrt{\ln n \ln\ln n})$, we can compute $M(n)$ in time $O(n^2/L(n)^{1/\sqrt{2}+o(1)})$.

For details, see our arXiv paper.

We have implemented the subquadratic algorithm for *evaluation* of $M(n)$. It should also be possible to *tabulate* all the values $M(k)$ for $1 \leqslant k \leqslant n$ in subquadratic time, but the algorithm is complicated and not yet implemented.

Although the time for evaluation/tabulation is subquadratic, the result is a little disappointing — we were hoping for something like Karatsuba's $O(n^{1.585})$, but we don't see how to obtain this. Since $L(n) = O(n^{\varepsilon})$ for all $\varepsilon > 0$, it's not clear how to evaluate $M(n)$ in time $O(n^{\lambda})$ for any $\lambda < 2$.

Monte Carlo computation

We can estimate $M(n)$ using two different Monte Carlo methods. Recall that

$$M(n) = \#S_n, \quad S_n = \{ij : 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant n\}.$$

Bernoulli trials

We can generate a random integer $x \in [1, n^2]$, and count a *success* if $x \in S_n$. Repeat several times and estimate

$$\frac{M(n)}{n^2} \approx \frac{\#\text{successes}}{\#\text{trials}}\,.$$

To check if $x \in S_n$ we need to find some of the divisors of x, which probably requires the prime factorisation of x. There is no obvious algorithm that is much more efficient than using the prime factors of x to construct divisors (though we can avoid factoring, as I'll explain later if time permits).

Monte Carlo computation - alternative method

There is another Monte Carlo algorithm, using what we call *product trials*.

Generate random integers $x, y \in [1, n]$. Count the number $\nu = \nu(xy)$ of ways that we can write $xy = ij$ with $i \leqslant n$, $j \leqslant n$. Repeat several times, and estimate

$$\frac{M(n)}{n^2} \approx \frac{\sum 1/\nu}{\#\text{trials}} \,.$$

This works because $z \in S_n$ is sampled at each trial with probability $\nu(z)/n^2$, so the weight $1/\nu(z)$ is necessary to give an unbiased estimate of $M(n)/n^2$.

To compute $\nu(xy)$ we need to find the divisors of xy.

Note that $x, y \leqslant n$, whereas for Bernoulli trials $x \leqslant n^2$, so the integers considered in product trials are generally smaller than those considered in Bernoulli trials.

Comparison of the Bernoulli and product methods

For Bernoulli trials, $p = M(n)/n^2$ is the probability of a success, and the distribution of the estimate for $M(n)/n^2$ after t trials has mean p/t, variance $p(1-p)/t \approx p/t$. (In what follows, remember that p is a function of n, and $p \to 0$ as $n \to \infty$.)

For product trials, the variance involves $\mathbb{E}(1/\nu^2)$, which we don't know. (In a computation, it can be estimated using the sample variance with Bessel's correction.)

The variance V_P after $t > 1$ trials of the product method is strictly smaller than the variance V_B after the same number of trials of the Bernoulli method (see next slide).

We find numerically that $V_P/V_B \in [1/8, 1/4]$ for $n \geqslant 3$ (at least up to the limits of our computations).

A more precise comparison of variances

Consider t fixed. V_B, V_P, p and ν are functions of n. Any asymptotic statements are as $n \to \infty$.

We have $tV_B = p(1-p)$ and

$$tV_P = \mathbb{E}[(\nu^{-1} - p)^2] = \mathbb{E}(\nu^{-2}) - p^2$$

but

$$\mathbb{E}(\nu^{-2}) \leqslant \mathbb{E}(\nu^{-1}) = p,$$

so $tV_P \leqslant p - p^2 = tV_B$. Thus $V_P \leqslant V_B$.

Numerical evidence suggests that V_P / V_B is bounded away from zero, so $V_P \asymp V_B$. However, we have not proved this.

For future reference, observe that
$\mathbb{E}(\nu^{-2}) = tV_P + p^2 \leqslant p \sim p - p^2 = tV_B$.

The function $T(n)$

In this table for $n = 7$ I have coloured the $T(n)$ entries above the diagonal that occur exactly twice in the full table. Clearly $T(7) = 16$, so $T(7)/M(7) = 0.64$.

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

In general, $T(n) \leqslant M(n)$ and
$T(n)/n^2 = \mathbb{P}\mathrm{rob}(\nu = 2) \leqslant 4\,\mathbb{E}(\nu^{-2}) \ll tV_B \sim p = M(n)/n^2$.

Thus $T(n) \asymp M(n) \implies V_P \asymp V_B$.

The numerical evidence suggests a very slow decrease in T/M, something like $T(n)/M(n) = O((\log\log n)^{-\alpha})$.

Numerical results

The ratios $T(n)/M(n)$ and $V_P(n)/V_B(n)$ are given in the table. (Monte Carlo estimates for $n > 100$, rms error $\lesssim$1 ulp.)

n	T/M	V_P/V_B	n	T/M	V_P/V_B
2	0.333	0.333	10^5	0.548	0.183
3	0.500	0.250	10^{10}	0.509	0.199
4	0.556	0.196	10^{20}	0.478	0.209
5	0.643	0.161	10^{30}	0.462	0.213
6	0.556	0.204	10^{40}	0.452	0.215
7	0.640	0.170	10^{50}	0.446	0.217
8	0.600	0.163	10^{60}	0.441	0.218
9	0.611	0.129	10^{70}	0.437	0.219
10	0.548	0.141	10^{80}	0.432	0.219
11	0.623	0.130	10^{90}	0.429	0.220
12	0.593	0.154	10^{100}	0.426	0.221
100	0.595	0.145	10^{200}	0.411	0.223

Theoretical results

The conjecture that $T(n) \asymp M(n)$ "almost" follows from results of Kevin Ford [Annals, 2008].

Ford says (via email, 18 May 2019):

> *"I suspect that Theorems* 4, 5 *remain valid for* $c = 0$, *but my method of proof breaks down there. In particular, this would imply that* $T(n) \gg M(n)$, *a positive proportion of entries occur just twice, and similarly a positive proportion occur exactly* $2k$ *times for any* $k \geqslant 1$. *As far as I know, nobody has worked this out; probably a good student project."*

If, on the other hand, the proportion of entries that occur exactly $2k$ times is $o(1)$ (for all $k \geqslant 1$), then it follows that $V_P(n) = o(V_B(n))$.

Proving this might be a project for a good student.

Further algorithmic considerations

When comparing the Bernoulli and product algorithms, comparing variances does not tell the whole story, because we also need to factor x (for Bernoulli trials) or xy (for product trials), and then find (some of) their divisors.

For large n, the most expensive step *appears* to be factoring, which is easier for product trials because the numbers involved are smaller (although the number of divisors is typically larger).

However, it turns out that factoring can be avoided!

Avoiding factoring large integers

We can avoid the factoring steps by generating random integers together with their factorisations, using algorithms due to Bach (1988) or Kalai (2003).

There is no time to describe these algorithms today. If you are interested, I suggest that you start with Kalai's paper for a description of his algorithm (since it is simpler than Bach's), then read Bach's paper, or the book *Algorithmic Number Theory* (by Bach and Shallit), for a description of Bach's algorithm.

When using Bach's or Kalai's algorithm, the most expensive step is primality checking. It is desirable to replace rigorous primality tests by the (much faster) Rabin-Miller probabilistic primality test, or a similar test that has a sufficiently small probability of error. An occasional error in primality testing will have only a small effect on the Monte Carlo estimates.

References

E. Bach, How to generate factored random numbers, *SIAM J. on Computing* **17** (1988), 179–193.

E. Bach and J. Shallit, *Algorithmic Number Theory*, Vol. 1, MIT Press, 1996.

R. P. Brent and H. T. Kung, The area-time complexity of binary multiplication, *J. ACM* **28** (1981), 521–534 & **29** (1982), 904.

R. P. Brent, C. Pomerance, D. Purdum and J. Webster, Algorithms for the multiplication problem, arXiv:1908.04251v1.

P. Erdős, Some remarks on number theory, *Riveon Lematematika* **9** (1955), 45–48 (Hebrew).

P. Erdős, An asymptotic inequality in the theory of numbers, *Vestnik Leningrad Univ.* **15** (1960), 41–49 (Russian).

K. Ford, The distribution of integers with a divisor in a given interval, *Annals of Math.* **168** (2008), 367–433.

H. A. Helfgott, An improved sieve of Eratosthenes, arXiv:1712.09130v2, 24 April 2018.

A. Kalai, Generating random factored numbers, easily, *J. Cryptology* **16** (2003), 287–289.

H. Maier, Primes in short intervals, *Michigan Math. J.* **32** (1985), 221–225.

T. Oliveira e Silva, Fast implementation of the segmented sieve of Eratosthenes, Dec. 28, 2015.

G. Tenenbaum, Sur la probabilité qu'un entier possède un diviseur dans un intervalle donné, *Compositio Math.* **51**(1984), 243–263 (French).