Computing $M(n)$

Jonathan Webster
Mathematics, Actuarial Science, and Statistics
Butler University
Indianapolis, IN 46208 USA
jewebste@butler.edu

David Purdum
Mathematics, Statistics, and Computer Science Major
Butler University
Indianapolis, IN 46208 USA
dpurdum@butler.edu

In Collaboration with Richard Brent (Australia National University, ACT) and Carl Pomerance (Dartmouth, New Hampshire)

Definitions and History

Let $M(n)$ be the function that counts the number of distinct numbers in an $n \times n$ multiplication table; that is,

$$M(n) = |\{ij : 1 \le i \le j \le n\}|.$$

The asymptotic study of $M(n)$ was initiated by Erdös (in Hebrew) [3]. He improved his own result (in Russian) [4] and Tenenbaum [7] further clarified (in French). The best currently known result is due to Ford (in English) [5]. He proved

$$M(n) \asymp \frac{n^2}{(\ln n)^c (\ln \ln n)^{3/2}} \text{ where } c = 1 - \frac{1 + \ln \ln 2}{\ln 2} = 0.086071\ldots.$$

Erdös, Tenenbaum, and Ford

Brent and Pomerance used algorithms of Bach [1] and Kalai [6] to provide Monte Carlo estimates of this function [2].

Brent, Pomerance, Bach, and Kalai

Exact Evaluation - Naive Algorithm

Algorithm 1: Computing $M(n)$

```
Initialize a bit vector A of length n^2 + 1 to 0.
for 1 ≤ i ≤ n do
    for i ≤ j ≤ n do
        Set A[ij] = 1
return Hamming weight of A
```

This algorithm is very similar to a standard sieve of Eratosthenes. One may process A in segments. Brent implemented a segmented version and evaluated $M(2^k - 1)$ for $k = 1, 2, \ldots, 25$.

Using this algorithm, evaluation is $O(n^2)$ in time and tabulation is $O(n^3)$ in time. The space requirement is $O(n^2)$.

Exact Evaluation - Incremental Algorithm

Assume we know $M(n-1)$. Consider the n products added when going from the $(n-1) \times (n-1)$ multiplication table to the $n \times n$ multiplication table: mn for $1 \le m \le n$. Some number of these products $D(n)$ are new distinct products and the remainder $\delta(n)$ of them were already in the $(n-1) \times (n-1)$ table. So,

$$M(n) = M(n-1) + D(n) = M(n-1) + (n - \delta(n)).$$

Computing $\delta(n)$

Assume we know the divisors of n. Let $n = gh$ and let $m = ij$. When can we express mn as a product of two numbers less than n? Then $ih \times jg$ will work iff $ih < n$ and $jg < n$ iff $i < g$ and $j < h$.

To compute $\delta(n)$, we need to count unique products ij with $0 < i < g$ and $0 < j < n/g$ for each divisor of n. The implementation does not duplicate work arising from overlapping rectangles of different divisors.

Algorithm 2: Computing $\delta(n)$

Input : $D = [[d_0 = 1, n], \ldots, [d_{\ell-1}, n/d_{\ell-1}]]$ contains the ordered divisors of n where $d_{\ell-1}$ is the largest divisor in $[1, \sqrt{n}]$.

```
Initialize counters i = 1 and k = 1 and a bit vector A of length n to 0.
for i < D[ℓ − 1][0] do
    if i == D[k][0] then
        Increment k
    for i ≤ j < D[k][1] do
        Set A[ij] = 1
return Hamming weight of A
```

This uses $O(n)$ space.

Computing $\delta(42)$

We can see the three rectangles corresponding to the divisor pairs $2 \cdot 21$, $3 \cdot 14$, and $6 \cdot 7$. The grey area corresponds to the products Algorithm 2 constructs.

6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Computing $\delta(63)$

We can see two rectangles corresponding to the divisor pairs $3 \cdot 21$ and $7 \cdot 9$. The grey area corresponds to the products Algorithm 2 constructs.

7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Computing $\delta(361)$

We can see the one rectangle corresponding to the divisor pairs $19 \cdot 19$. Note, $\delta(361) = M(18)$.

19	19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361
18	18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342
17	17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323
16	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304
15	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285
14	14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266
13	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247
12	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228
11	11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Run-Time Analysis

There are two straightforward bounds on the run-time of Algorithm 2.

- $O(n \ln n)$: Note that the products constructed are less than n and so lie under a hyperbola. This bound over counts by counting numbers which should not be counted.
- $O(n\tau(n))$: Let $\tau(n)$ count the number of divisors of n. Then one rectangle (of area less than n) is constructed for each divisor of n. This over counts because area that lies in the intersections of rectangles is counted more than once.

Each of these bounds may be used to get a run-time for tabulating $M(n)$ as $O(n^2 \ln n)$. We can do better. Let $\tau(n; y, z)$ be the number of divisors d of n which satisfy $y < d \le z$ and

$$\tau^+(n) = |\{k \in \mathbb{Z} : \tau(n, 2^k, 2^{k+1}) \ge 1\}|.$$

Theorem. *Algorithm 2 computes $\delta(n)$ in time $O(n\tau^+(n))$.*

Proof. Let the area be $\mathcal{A}$. For each k, consider all the divisors of n in the interval $(2^k, 2^{k+1}]$. They all have the same bottom left corner, namely, the origin, and shapes range from $2^k \times n/2^k$ to $2^{k+1} \times n/2^{k+1}$. Hence they are all enclosed by a rectangle of shape $2^{k+1} \times n/2^k$ which has area $2n$. Thus we get an upper bound $\mathcal{A} \le 2n\tau^+(n)$. □

Theorem. *Algorithm 2 tabulates $M(n)$ in time*

$$O\left(\frac{n^2 (\ln n)^{1-c}}{(\ln \ln n)^{3/2}}\right) \text{ where } c = 1 - \frac{1 + \ln \ln 2}{\ln 2} = 0.086071\ldots.$$

Proof. Compute $M(n)$ by evaluating $\delta(k)$ for $1 \le k \le n$. The run-time is

$$O\left(\sum_{k \le n} k\tau^+(k)\right) = O\left(n^2 \left(\frac{1}{n} \sum_{k \le n} \tau^+(k)\right)\right) = O\left(\frac{n^2 (\ln n)^{1-c}}{(\ln \ln n)^{3/2}}\right).$$

Corollary 5 of [5] gives the last equality. □

Results

k	$M(2^k - 1)$
26	830751566970326
27	3288580294256952
28	13023772682665848
29	51598848881797343

Algorithm 2 was independently implemented by David Purdum (C++) and Richard Brent (C).

References

[1] Eric Bach, *How to generate factored random numbers*, SIAM J. Comput. **17** (1988), no. 2, 179–193, Special issue on cryptography. MR 935336

[2] R. P. Brent and C. Pomerance, *Algorithms for the multiplication table problem*, http://maths-people.anu.edu.au/~brent/pd/multiplication-CARMA.pdf.

[3] Paul Erdös, *Some remarks on number theory*, Riveon Lematematika **9** (1955), 45–48.

[4] _____, *An asymptotic inequality in the theory of numbers*, Vestnik Leningrad. Univ. **15** (1960), no. 13, 41–49.

[5] Kevin Ford, *The distribution of integers with a divisor in a given interval*, Ann. of Math. (2) **168** (2008), no. 2, 367–433.

[6] Adam Kalai, *Generating random factored numbers, easily*, J. Cryptology **16** (2003), no. 4, 287–289. MR 2002046

[7] G. Tenenbaum, *Sur la probabilité qu'un entier possède un diviseur dans un intervalle donné*, Seminar on number theory, Paris 1981–82 (Paris, 1981/1982), Progr. Math., vol. 38, Birkhäuser Boston, Boston, MA, 1983, pp. 303–312.