# The curve of matrix multiplication schemes

Warren D. Smith, Nov.2023. warren.wds@gmail.com

**ABSTRACT.** Various methods have been devised to multiply N×N matrices using $O(N^E)$ arithmetic operations when N→∞, for various exponents E with 2<E≤3. However, in practice, the schemes with least known E are *not* the best, because they only start to win for infeasibly large N. Prior literature has unhealthily been fixated on E alone, i.e. on the asymptotic large-N performance alone. To address that, we propose investigating, for each method, not merely its E, but also its "breakeven N," meaning the least N causing that method to use fewer than the obvious algorithm's $N^3$ bilinear multiplications [or fewer than Strassen's $O(N^{2.807355})$ scheme]. The set of (E,logN) datapoints then form a subset of the infinite rectangle (2,3]×[0,∞). Part of that rectangle is filled with datapoints, while another part contains none. What is of interest is the *curve* delineating the boundary between those two regions.

Along the way we also provide the best available review (with new results and a numerical table) of bounds on the Salem-Spencer function: the cardinality of the largest subset of {1,2,3,...,X} free of 3-term arithmetic progressions.

## Notation

The "(a,b,c) matrix multiplication problem" shall mean multiplying a×b and b×c matrices to obtain an a×c matrix. We shall mainly be interested in the (N,N,N), i.e. square-matrices, case. Any M-multiplication bilinear algorithm to solve (a,b,c) yields as a consequence bilinear algorithms for (N,N,N) with $O(N^E)$, where E=3log(M)/log(abc), arithmetic operations. Rk[T] shall mean the minimum number of multiplications in a bilinear algorithm for solving problem T. And $Rk_h[T]$ shall mean the same thing, but for "APA algorithms of degree=h," and $\underline{Rk}(T)=Rk_\infty[T]$ shall mean in the case where unboundedly large h is permitted. It is known ("hexality") that (a,b,c), (a,c,b), (b,a,c), (b,c,a), (c,a,b), and (c,b,a) all have the same Rk[T]. Obviously Rk[(a,b,c)] is an increasing function of a,b, and c (proof: consider "zero padding"). We shall use T=U⊕V to denote the task T consisting of solving *both* problem U and (apparently wholy independent) problem V. And T=U⊗V denotes the "tensor product" task where each element of task U is replaced by an instance of task V; here note (aA,bB,cC)=(a,b,c)⊗(A,B,C) because each entry of the a×b, b×c, and a×c matrices is replaced by a block, the respective block sizes being A×B, B×C, and A×C. It is known that ⊗ is associative and together with ⊕ satisfies the distributive law, Rk[U⊗V]≤Rk[U]·Rk[V] and Rk[U]≤Rk[U⊕V]≤Rk[U]+Rk[V] and Rk[(a+A,b,c)]≤Rk[(a,b,c)]+Rk[(A,b,c)]. And correspondingly for the $Rk_h$ and $\underline{Rk}$ variations, albeit with $Rk_{max(j,h)}[U\oplus V]\le Rk_j[U]+Rk_h[V]$ and $Rk_{j+h}[U\otimes V]\le Rk_j[U]\cdot Rk_h[V]$. Finally U⊗U⊗...⊗U with p letters "U" is the "$p^{th}$ tensor power" of U, written $U^{\otimes p}$.

## Strassen versus Obvious

The obvious matrix multiplication method uses $M_{obv}(N)=N^3$ elementwise multiplication and $A_{obv}(N)=(N-1)N^2$ addition ops. In 1969, V.Strassen invented a formula to multiply 2×2 matrices *not* via the obvious (8×, 4+) scheme, but rather (7×, 18±), which Winograd 1971 improved to (7×, 15±). Strassen's "7" was shown to be best possible by both Hopcroft & Kerr and Winograd independently. Probert 1976 proved that Winograd's "15" was best possible if we are using 7 muls. However, Karstadt & Schwartz 2020 pointed out a way to do it in (7×, 12±) if the basis is changed (also showing "12" is best possible) provided the basis-change operations are not counted as part of the "cost."

Importantly, both the obvious, Strassen's, Winograd's, and Karstadt & Schwartz's formulas work even if the 2×2 matrices' entries are members of an arbitrary possibly-*noncommutative* ring. Therefore, we can recurse, i.e. can multiply 2N×2N matrices (even-sized) via 15 additions and 7 multiplications of N×N subblocks. Odd-sized matrices can be handled by treating their last row and last column conventionally, then handling the four (N-1)/2×(N-1)/2 remaining blocks by Strassen recursion. If T(N) denotes the runtime (e.g. op-count) to multiply N×N matrices, then we have $T(2N)\le 7T(N)+O(N^2)$ and $T(2N+1)\le 7T(N)+O(N^2)$ where for all sufficiently-small N we may use the obvious scheme, i.e. if $1\le N\le N_0$ then use $T(N)=N^3$ muls plus $(N-1)N^2$ adds. (Choose the threshhold $N_0$ to optimize performance.) The solution is $T(N)=O(N^S)$ where $\mathbf{S=\log_2 7\approx 2.807354922}$ is the **Strassen exponent**.

Unfortunately by using O-notation, we've *hidden* the constant factor. If we let $A_{str}(N)$ denote the number of addition/subtraction ops, and $\mathbf{M_{str}(N)}$ denote the number of multiplications, then $A_{str}(2N)=7A_{str}(N)+15N^2$, $M_{str}(2N)=7M_{str}(N)$, $A_{str}(2N+1)=7A_{str}(N)+15N^2+(12N+2)N$, $M_{str}(2N+1)=7M_{str}(N)+(12N+6)N+1$, except that $A_{str}(N)=A_{obv}(N)$ and $M_{str}(N)=M_{obv}(N)$ if $N\le N_0$. The simplest (but perhaps stupid) choice is to use $N_0=1$. I'll call that "**Str1**." These recurrences may equivalently be written in "top down" rather than "bottom up" style:

$$M_{str}(N) = 7M_{str}(\lfloor N/2\rfloor) + (N \bmod 2)(3[N-1]N+1), \qquad A_{str}(N) = 7A_{str}(\lfloor N/2\rfloor) + 15\lfloor N/2\rfloor^2 + (N \bmod 2)(3N-2)(N-1)$$

if $N>N_0$, with the base cases when $N_0=1$ being $M_{str1}(1)=1$ and $A_{str1}(1)=0$. These show $1\le N^{-S}M_{str1}(N)\le 47/30\approx 1.5666$ and $2.7666\approx 83/30\le N^{-S}A_{str1}(N)\le 5$ and we get

| N | $M_{obv}(N)$ | $A_{obv}(N)$ | $M_{str1}(N)$ | $A_{str1}(N)$ | N | $M_{obv}(N)$ | $A_{obv}(N)$ | $M_{str1}(N)$ | $A_{str1}(N)$ |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 8 | 4 | 7 | 15 | 3 | 27 | 18 | 26 | 29 |
| 4 | 64 | 48 | 49 | 165 | 7 | 343 | 294 | 309 | 452 |
| 8 | 512 | 448 | 343 | 1395 | 15 | 3375 | 3150 | 2794 | 4501 |
| 16 | 4096 | 3840 | 2401 | 10725 | 31 | 29791 | 28830 | 22349 | 37612 |
| 32 | 32768 | 31744 | 16807 | 78915 | 63 | 250047 | 246078 | 168162 | 289293 |
| 64 | 262144 | 258048 | 117649 | 567765 | 127 | 2048383 | 2032254 | 1225141 | 2132340 |
| 128 | 2097152 | 2080768 | 823543 | 4035795 | 255 | 16581375 | 16516350 | 8770298 | 15362117 |
| 256 | 16777216 | 16711680 | 5764801 | 28496325 | 511 | 133432831 | 133171710 | 62173917 | 109291004 |
| 512 | 134217728 | 133955584 | 40353607 | 200457315 | 1023 | 1070599167 | 1069552638 | 438353938 | 772088317 |
| 1024 | 1073741824 | 1072693248 | 282475249 | 1407133365 | 2047 | 8577357823 | 8573167614 | 3081042053 | 5432876548 |
| 2048 | 8589934592 | 8585740288 | 1977326743 | 9865662195 | 4095 | 68669157375 | 68652388350 | 21617589162 | 38143275573 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4096 | 68719476736 | 68702699520 | 13841287201 | 69122549925 | 8191 | 549554511871 | 549487419390 | 151524377005 | 267455700876 |
| 8192 | 549755813888 | 549688705024 | 96889010407 | 484109507715 | 16383 | 4397241253887 | 4396972851198 | 1061475797954 | 187400141950 |
| 16384 | 4398046511104 | 4397778075648 | 678223072849 | 3389773186965 | 32767 | 35181150961663 | 35180077285374 | 7433551516245 | 131252568418 |
| 32768 | 35184372088832 | 35183298347008 | 47475615099943 | 23732438840595 | 65535 | 281462092005375 | 281457797169150 | 52047744925786 | 919057872184 |

Therefore on a machine on which multiplication is >11 times more expensive than addition, Str1 would beat Obvious for every N≥2. But even on a machine on which multiplication and addition took equal time (and even using $N_0$=1) the table shows Str1 would beat Obvious when N=512 and N=63, and hence for every N≥512. (And even in a ridiculous world where addition was arbitrarily *more* expensive than multiplication, Str1 still would win when N=8192 and N=255 and hence for every N≥8192.) If however we note that even on such an equal-time machine multiplying N×N *matrices* by the obvious method *is* >11 times more expensive than adding those matrices for each N>6, then we see that Strassen-Winograd using $N_0$=13, **"Str13," should beat Obvious for every N≥14** based on total (add & mul) op-count alone (and tie it for 1≤N≤13). This has ignored the fact Strassen-Winograd performs extra data copying and temporary storage. In addition to the $3N^2$ entries of the 2 input and 1 output matrices, $(2/3)N^2$ *extra* words of storage are required if Boyer et al 2009's memory-efficient scheduling (their "table 1" due to Douglas et al 1994, or "table 9") are employed; Boyer "algorithm 2" gives a schedule using no extra storage at all ("in place") but increasing the constant factor in Strassen's arithmetic-op count by 20%; neither extra storage nor any constant factor op-count increase are needed if we permit overwriting both input matrices (their "algorithm 1").

The Karstadt-Schwartz (7×, 12±) changed-basis verson of Strassen can actually be made to work to reduce the total arithmetic-op count by a factor of 1.2 versus Strassen-Winograd when N→∞. They claim it experimentally outperforms Strassen-Winograd by about 12% when N=32768.

What happens on **real machines**? I first began programming on 8-bit microprocessors such as the Zilog Z80, Mostek 6502, Intel 8051, and RCA 1802 which had no hardware multiplication instruction. If you wanted to multiply, you needed to program multiplication using shifts and adds. (The Z80 was first sold in 1976 with clock speed 2.5 MHz, contained 8500 transistors, ran on a single 5-volt power supply thanks to using NMOS technology, and on average code would execute about 1 instruction every 7 clock cycles. Although the Z80 connected to external memory via an 8-bit-wide bus, internally some of its instructions could operate on 16-bit-wide data.) I found on the web some 64-bit unsigned integer multiply (128-bit output) routines for the Z80 claiming average runtime 8721 cycles; and other code (codelength 45 bytes) to add or subtract 64-bits words in 294 cycles. So the multiply/add time-ratio for 64-bit integers on the Z80 was about 8721/294≈29.66. Since this far exceeds 11, **for matrices of 64-bit numbers on the Z80, Strassen would have been superior to Obvious for every matrix size N≥2**.

In 1979, the Motorola 68000 processor came out. It had 68000 transistors, still used 5-volt NMOS, had clock frequencies 4-17 MHz, 16-bit bus but many 32-wide internal instructions, and on average code performed about 1 instruction every 5.7 clock cycles. (The "68SEC000," a CMOS version of the original 68000, as of year 2023 is still in production from Freescale Semiconductor Inc. with clock speeds 10-20 MHz and supply voltage 3.5-5V. The "eZ80," a faster, CMOS version of the Z80, now enhanced with many 24-bit-wide instructions, was introduced in 2001 and still was produced by Zilog Inc. as of year 2021. Its clock rates are up to 50 MHz and thanks to redesigned internal pipelines executes instructions about 3 times faster than the original design would at equal clock rates.) Unlike the Z80, the 68000 provided a multiplication instruction for unsigned 32-bit words (64-bit product) running in 38+2n clock cycles where n equals the number of nonzero-bits in the operand, e.g. typically 70 cycles. Meanwhile adding took 4 or 8 cycles. But if you wanted to multiply *64-bit*-wide words, then multiplication was going to be at least 22 times more expensive than addition, so again **Strassen on the 68000 would be superior to Obvious for every N≥2**.

Today (year 2023) that is no longer the case. Modern CPUs use 1.2-volt CMOS, sometimes have over $10^{11}$ transistors on chip, operate at clock rates 1-9 GHz, and perform 64-bit-wide ops including both integer and floating point multiplication using maximally-parallelized hardware, with *comparable* speeds for addition, multiplication, and just copying data between different memory locations. They have fancy pipelining, branch prediction, and multicore parallelism schemes enabling average instruction rates far exceeding clock rate. Data copying was considered an almost neglectible cost – much cheaper than ± or × – on old machines like the Z80, but on year-2023 machines can be quite expensive. Furthermore, some modern machines have "**vectorization**" capability permitting extremely fast computation of vector inner products – i.e. the inner loop of the obvious matrix-multiplication method – providing an artificial advantage for "Obvious."

All this has caused claims that the breakeven N for Strassen can be as high as "several thousand" for some a priori reasonable-looking software on some modern hardware – quite an astounding change from the value "2" valid when I began programming in the late 1970s!

But I have trouble believing that any reasonable hardware and software at all resembling today's Von Neumann machines will ever disagree that Strassen beats Obvious for all N≥4000. And, if the matrix entries are sufficiently-wide **multiprecision** numbers (e.g. ≥10 words wide) then even on today's machines Strassen should still beat Obvious for every N≥2. That's because for wide multiprecision numbers, multiplication *is* arbitrarily more expensive than either addition, subtraction, or scalings by rational constants with bounded numerators and denominators, since all the latter have linear-time algorithms, while none are known (and presumably none exist) for integer multiplication.

## The curve. Our definition of "breakeven N."

**Definitions:** If some scheme for multiplying N×N matrices has op-count T(N) which obeys $\lim_{N\to\infty} \log(T(N))/\log(N)=E$, then I say it has **exponent E**. Aside from Strassen and Obvious, we are only going to be considering schemes with $E<S\equiv\log_2 7\approx 2.807354922$. The least N>1 causing the scheme to employ $<N^3$ bilinear multiplications, is its **breakeven N**. For "mass production" schemes that do not just multiply *one* pair of N×N matrices, but in fact K independently specifiable such pairs, the breakeven N is the least N>1 causing the scheme to employ $<N^3K$ bilinear multiplications.

My definition of "breakeven N" is the simplest. But it can be criticized for reasons we've already discussed causing the "true" breakeven N to perhaps be up to a factor 2000 greater than my definition (depending on hardware & software). Also my definition is subject to abuse by "cheaters." (Don't do that.) Further, if you wanted for the mul-count not merely to go below $N^3$ but in fact to *beat it by a factor of 2*, then you'd need to multiply our breakeven N by a factor up to $2^{1/(3-E)}$, which is <37 for schemes with E<2.807355.

Also interesting is the breakeven N versus Strassen rather than versus Obvious. I'll call those $\mathbf{N_{str}}$ and $\mathbf{N_{obv}}$.

It also is interesting to consider your algorithm's breakeven N versus the *best* matrix multiplication algorithm with greater E than yours... that being, arguably, the N that matters most. But the trouble with that is that I *do not know* the best ones. Therefore, I'm mainly going to stick with Obvious and Strassen.

It seems nicest to plot the resulting set of (E,N) datapoints on **semilog paper**, i.e. instead plot $(E,\log_2 N)$. The main goal of the present work is simply to produce that plot.

| Exponent E | Breakeven $N_{obv}$ | $Log_2N_{obv}$ | Breakeven $N_{str}$ | $Log_2N_{str}$ | Comment |
|---|---|---|---|---|---|
| 3 | 1 | 0 | | | The Obvious matrix multiplication method |
| 2.807354922 | 2 | 1 | 2 | 1 | Strassen 1969 (Winograd variant) 7-mul formula for (2,2,2), all coeffs. in {0,±1}. |
| 2.795668800 | 12 | 3.5850 | 12 | 3.5850 | 1040-mul formula from Rk[(2,4,4)]≤26 and Rk[(6,3,3)]≤40 |
| 2.792341873 | 18 | 4.1699 | 18 | 4.1699 | 3200-mul formula from Rk[(6,3,3)]≤40 |
| 2.790137008 | 22 | 4.4594 | 22 | 4.4594 | 5566-mul formula by Drevet, Nazrul Islam, Schost 2011 |
| 2.785876483 | 24 | 4.5850 | 24 | 4.5850 | 7000-mul formula by Drevet, Nazrul Islam, Schost 2011 |
| 2.782679679 | 26 | 4.7004 | 26 | 4.7004 | 8658-mul formula by Drevet, Nazrul Islam, Schost 2011 |
| 2.780276441 | 28 | 4.8074 | 28 | 4.8074 | 10556-mul formula by Drevet, Nazrul Islam, Schost 2011 |
| 2.780141891 | 48 | 5.5850 | 48 | 5.5850 | Pan 1980: $(2n^3+27n^2-2n)/6$-mul exact formula for (n,n,n) with n=even, here when n=48. Hadas & Schwartz 2023 have a redo of Pan 1980 which they claim will multiply N×N matrices with total op-count $[2+o(1)]N^E$ with the same exponent E=ln(47216)/ln(48)=2.78014 as Pan; and they say they can beat Strassen's total op-count when N≈13800. |
| 2.779885224 | 2985984 | 21.5098 | $7.032\times10^{60}$ | 283.2120 | 10-mul APA formula by Bini et al 1979 for (2,2,3) with degree=1, arising from tiling 2 copies of a 5-mul $APA_1$ formula for (2,2,2) with 1 input zeroed. |
| 2.774299980 | 54 | 5.7549 | 54 | 5.7549 | Smirnov 2013: 40-mul exact formula for (3,3,6), all coefficients in ±{0,1/8,1}. |
| 2.773371017 | 44 | 5.4594 | 44 | 5.4594 | Pan 1982: $(4n^3+45n^2+128n+108)/12$-mul exact formula for (n,n,n) with n=even, here when n=44. Hadas & Schwartz 2023 have a redo of Pan 1982 which they claim will multiply N×N matrices with total op-count $[8.082+o(1)]N^E$ with the same exponent E=ln(36133)/ln(44)≈2.77337 as Pan. |
| 2.739153525 | 1889568 | 20.8496 | $2.4088\times10^{31}$ | 104.2481 | Smirnov 14-mul APA formula for (3,2,3) with degree=2. |
| 2.728435621 | ? | ? | ? | ? | Smith 18-mul APA formula for (2,3,4) if valid (unknown degree). |
| 2.726833028 | 43046721 | 25.3594 | $3.0903\times10^{31}$ | 104.6075 | Smirnov 20-mul APA formula for (3,3,3) with degree=6. |
| 2.547992912 | 262144 | 18 | $2.749\times10^{11}$ | 38 | Schönhage 1981: lemma 6.1 with k=n=4 via his ASI. |
| 2.478495141 | 1220703125 | 30.1851 | $1.1921\times10^{16}$ | 53.4043 | My simplified version of a Strassen 1987 "laser" method. |
| 2.403632261 | ≤68719476736 | ≤36 | $\le3.7779\times10^{22}$ | ≤75 | "Baby" Coppersmith-Winograd 1990, their §6 |
| 2.3871900 | 3656158440062976 | 51.6993 | $\le6.189\times10^{28}$ | ≤95.6436 | "Toddler" Coppersmith-Winograd 1990, their §7 |
| 2.3754770 | ? | ? | ? | ? | "Monster" Coppersmith-Winograd 1990, their §8 |

## APA (arbitrary precision approximate) formulae

"APA formulae" arise when the coefficients in a formula for multiplying matrices AB are *not* ordinary numbers like ±1, but rather *polynomials* in some variable x; and the matrix product C=AB arises as $C=x^{-h}$formula(A,B)+error where error=O(x) in the *limit* x→0. (Such a formula is said to be "approximate with degree=h." For any specific value of x, APA formulae will in general deliver wrong answers.) APA formulae can be converted to ordinary exact formulae by taking appropriate linear combinations of the APA(x) formulae for h+1 different values of x.

In particular, if we let x be h+1 different complex numbers forming a regular (h+1)-gon centered at the origin, then the required "linear combination" is simply the "average." If our matrices had complex-number entries, this yields an exact formula in work (for us measured as number of bilinear multiplications) equivalent to h+1 specified-x invocations of the APA formula.

For multiplicand matrices A,B with *real* entries, we do not need all h+1 roots of unity, only ⌊(h+1)/2⌋, because those in the lower halfplane yield results which are the complex conjugates of the results arising from roots we'd already handled in the upper halfplane. But (except when h≤1) this trick is *not* good enough to reduce the cost to equivalent to h+1 specified-x invocations of the APA formula, because a complex multiplication costs 3, not 2, real multiplications. So the cost-factor exceeds h+1 by 50%.

Nevertheless, one can achieve that goal: use h+1 specified real values of x, although unfortunately the unequal both-signed weights now required in the linear combination increase numerical roundoff errors. Any h+1 distinct values $x_k$ work – due to the known formula for Vandermonde determinants being nonzero, there is always a unique linear combination of the $P(x_k)$ guaranteed to equal P(0). If the polynomial P(x) had rational coefficients and the $x_k$ are rational, then the coefficients in the linear combination will also be rationals.

**Examples:** Let L(x) be a linear, Q(x) a quadratic, C(x) a cubic, F(x) a fourth-degree polynomial, and Z(x) fifth-degree. Then here is a list of two kinds ("**geometric**" and "**arithmetic**") of formulae for P(0):
L(0) = [2L(½)+L(-1)]/3 = [L(-1)+L(1)]/2.
Q(0) = [8Q(½)+2Q(-1)-Q(2)]/9 = [Q(-1)+3Q(1)-Q(2)]/3.
C(0) = [64C(-¼)+24C(½)-6C(-1)-C(2)]/81 = [-C(-2)+4C(-1)+4C(1)-C(2)]/6.
F(0) = [1024F(-¼)+320F(½)-120F(-1)-10F(2)+F(-4)]/1215 = [-F(-2)+5F(-1)+10F(1)-5F(2)+F(3)]/10.
Z(0) = [32768Z(-¼)+11264Z(½)-3520Z(-1)-440Z(2)+22Z(-4)+Z(8)]/40095 = [Z(-3)-6Z(-2)+15Z(-1)+15Z(1)-6Z(2)+Z(3)]/20.

These lists of formulae exhibit some patterns. Those patterns may be proven valid by the theory of Lagrange interpolation, and alternatively by induction on the degree. The sequence of lefthand formulas [which I am calling "geometric" because their $x_k$ are proportional to $(-2)^k$] has the sequence of largest coefficient $|\text{ratios}|=2^{(h+1)h/2}$, exhibiting superexponential growth.

The sequence of righthand formulas, which I call "arithmetic" since their $x_k$ are the h+1 nonzero integers with least absolute values, sorted into increasing order starting from -⌊(h+1)/2⌋), are expressible in closed form. The coefficient-numerators for h=0,1,2,...,8 arise from this all-integer **number-triangle**

```
                                          1
                                       1     1
                                    1     3    −1
                                −1     4     4    −1
                             −1     5    10    −5     1
                           1    −6    15    15    −6     1
                        1    −7    21    35   −21     7    −1
                    −1     8   −28    56    56   −28     8    −1
                 −1     9   −36    84   126   −84    36    −9     1
              1   −10    45  −120   210   210  −120    45   −10     1
           1   −11    55  −165   330   462  −330   165   −55    11    −1
       −1    12   −66   220  −495   792   792  −495   220   −66    12    −1
    −1    13   −78   286  −715  1287  1716 −1287   715  −286    78   −13     1
```

where note that every number (if we ignore its sign) is the sum of the two above it (as in "Pascal's triangle") *except* for the numbers in the middle column. Each column of the triangle has constant sign. Rows containing an even number of entries are palindromic, aka even-symmetric. Rows containing an odd number of entries have odd symmetry if their (always positive) central entry is ignored; their row-sum therefore equals their central entry. Those odd-cardinality rows have rightmost entry 1, -1, 1, -1, 1, -1, ... alternating sign with period=2. (Also true about the even-cardinality rows.) The sequence 1,2,3,6,10,20,35,70,126... of common denominators are the row-sums (now *not* ignoring signs!) and arise from the formula binomial(n, $\lfloor n/2 \rfloor$) where n=h+1 is the cardinality of that row. These facts are enough to completely specify the entire infinite triangle. Since binomial(n, $\lfloor n/2 \rfloor$) is upper-bounded by $2^h$, this arithmetic class of formulas has largest coefficient |ratio| growing merely exponentially(h) – more desirable from the standpoint of numerical precision than the geometric formulas' *super*exponential growth.

What is the **best** choice of the $x_k$ for k=0,1,2,...,h? If h is odd, there is reason to argue that the answer is $x_k=\cos(k\pi/h)$. This choice maximizes the (h+1)×(h+1) Vandermonde |determinant| for h+1 points on the real interval [-1,1], or equivalently (by considering the logarithm of that |determinant|) minimizes the potential energy of h+1 mutually-repelling unit electrostatic charges (logarithmic potential) located on that interval. However, if h>3 most of these $x_k$ are irrational, which would be rather sad if the entries of your matrices happened to be exact rational numbers.

All that works for matrices whose entries are integer, rational, real, or complex, which are the most important cases in practice. However, trouble can sometimes arise for APA formulas if the matrix-entries instead are elements of the wrong **finite field** or **ring**. E.g. the first few geometric and arithmetic formulas we tabulated involve *division* by 2, 3, 6, or 81. In such rings as "the integers mod 6" which contain the wrong "zero-divisors" (here 2 and 3) all four of those divisions are *forbidden*! Also the latter three divisions are forbidden in the field of "integers mod 3." I shall not delve into that.

An M-multiplication degree-h APA formula for multiplying N×N matrices ($2\le M\lesseqgtr N^S$) has breakevens

$$N_{obv} = N^{\lceil k \rceil} \quad \text{where} \quad k = \ln(N^{-3}M)^{-1}\ \mathrm{LambertW}_{-1}(M^{1/h}N^{-3/h}h^{-1}\ln(N^3M^{-1})) - h^{-1}$$

and

$$N_{str} \le N^{\lceil k \rceil} \quad \text{where} \quad k = \ln(N^{-S}M)^{-1}\ \mathrm{LambertW}_{-1}(M^{1/h}N^{-S/h}h^{-1}\ln(N^SM^{-1})) - h^{-1}.$$

These formulas arise from solving $kh+1=(N^3/M)^k$ or $kh+1=(N^S/M)^k$ for k≥1. The $\mathrm{LambertW}_{-1}(x)$ function is the real value of y with y<-1 obeying $e^y y=x$. It is defined for $-e^{-1}\le x<0$. Actually that $N_{str}$ formula can yield overestimates in cases where round-to-integer effects are unfavorable for Strassen, which is why I wrote "≤," but the overestimation factors usually are not large.

**Example: Bini et al 1979**'s 10-mul degree-1 APA formula for (2,2,3) matrix multiplication yields an M=1000 multiplication APA formula with degree h=3 for multiplying 12×12 matrices: $Rk_3[(12,12,12)]\le 1000$. Compare the best known (as of year 2023) exact formula, which uses 1040 muls, and $12^3=1728$ for the obvious method. We find $k_{obv}=-3^{-1}\ln(6/5)^{-1}\mathrm{LambertW}_{-1}(-6^{-1}5\ln(6/5))-3^{-1}\approx 5.10453$ solves $3k+1=(6/5)^{3k}$ with k≥1. Hence $\lceil k_{obv} \rceil=6$ and Bini's breakeven $N_{obv}=12^6=2985984$. Similarly $k_{str}\approx 80.4302$ from solving $3k+1=\exp([\ln(12)\ln(7)/\ln(2)-\ln(1000)]k)$ for k>1. Hence $\lceil k_{str} \rceil=81$ suggesting Bini's breakeven $N_{str}\le 12^{81}\approx 2.592\times 10^{87}$.

That is not exact; it is an overestimate. To verify that we compute $M_{str1}(12^{81})\approx 3.36883\times 10^{245}$ and compare that to the number of bilinear multiplications $244\times 1000^{81}=2.44\times 10^{245}$ used by Bini to multiply N×N matrices with $N=12^{81}$. If $N=12^{80}$ then $M_{str1}(12^{80})\approx 3.44444\times 10^{242}$ while Bini uses $241\times 1000^{80}=2.41\times 10^{242}$. If $N=12^{79}$ then $M_{str1}(12^{79})\approx 2.764\times 10^{239}$ while Bini uses $2.38\times 10^{239}$. If $N=12^{78}$ then $M_{str1}(12^{78})\approx 2.667\times 10^{236}$ while Bini uses $2350\times 10^{236}$. So the exact value of $N_{str}$ for Bini is $12^{79}\approx 7.032\times 10^{60}$ with $\log_2 N_{str}\approx 283.2120$.

**Example: Smirnov 2013**'s 14-mul degree-2 APA formula for (3,2,3) matrix multiplication yields an $M=14^3=2744$ multiplication APA formula with degree h=6 for multiplying 18×18 matrices. (The least known mul-count for an exact formula is 3200; the obvious method uses $18^3=5832$.) We find $k_{obv}=-3^{-1}\ln(9/7)^{-1}\mathrm{LambertW}_{-1}(-6^{-1}7^{1/2}\ln(9/7))-6^{-1}\approx 4.38727$ solves $6k+1=(9/7)^{3k}$ with k≥1. Hence $\lceil k_{obv} \rceil=5$ and the breakeven $N_{obv}=18^5=1889568$. Similarly $k_{str}\approx 25.5645$. Hence $\lceil k_{str} \rceil=26$ and the breakeven $N_{str}\le 18^{26}\approx 4.336\times 10^{32}$. Again this is an overestimate, and exact comparisons of $M_{str1}(N)$ and Smirnov mul-counts show that the exact $N_{str}$, i.e. the least N such that Smirnov's mul-count is below Strassen's, is $18^{25}\approx 2.4088\times 10^{31}$ with $\log_2 N_{str}\approx 104.2481$.

**Example: Smirnov 2013** also found a 20-mul APA formula with degree h=6 for multiplying 3×3 matrices. That compares with 23 muls for the most efficient known exact formula (found by Laderman, Sykora, and various others) and 27 for the obvious method. We find $k_{obv}\approx 15.0397$ solves $6k+1=(27/20)^k$ with k≥1. Hence $\lceil k_{obv} \rceil=16$ and the breakeven $N_{obv}=3^{16}=43046721$. Similarly $k_{str}\approx 67.9766$. Hence $\lceil k_{str} \rceil=68$ and the breakeven $N_{str}\le 3^{68}\approx 2.781\times 10^{32}$. Again this is an overestimate, and exactly comparing $M_{str1}(N)$ versus Smirnov's mul-counts show that the exact $N_{str}$ is $3^{66}\approx 3.0903\times 10^{31}$ with $\log_2 N_{str}\approx 104.6075$.

## Schönhage 1981's "asymptotic sum inequality" (ASI) and his specific result E≈2.547993

Schönhage in his lemma 6.1 stated an APA formula proving $Rk_2[(k,1,n)\oplus(1,m,1)]\le kn+1$ where m=(k-1)(n-1) and k,n≥2.

In contrast, Rk[(k,1,n)⊕(1,m,1)]=kn+m=Rk[(k,1,n)]+Rk[(1,m,1)]. More generally Strassen has stated the still-open "**additivity conjecture**" that Rk[U⊕V⊕...⊕W]=Rk[U]+Rk[V]+...+Rk[W] always holds. I'm skeptical – but if true, that would imply that improvements like Schönhage's lemma 6.1 can *only* be obtained using APA algorithms.

Schönhage's "asymptotic sum inequality" (ASI – sometimes more-stupidly called his "tau theorem") shows that his lemma 6.1 yields a matrix multiplication algorithm with exponent≤E≈2.54799291220440756106, where this E solves $16^E+9^E=17^3$. This is the least exponent obtainable from Strassen's lemma 6.1 and the ASI and arises from k=n=4, i.e. from Schönhage's 17-mul APA algorithm for simultaneously performing (4,1,4) and (1,9,1) rectangular matrix multiplications approximately with accuracy degree=2.

More generally, Schönhage's ASI claims that if $\underline{Rk}[(a_1,b_1,c_1)\oplus(a_2,b_2,c_2)\oplus \dots \oplus(a_p,b_p,c_p)]\le M$, then the exponent E of matrix multiplication is upper-bounded by the solution E of $\sum_{1\le j\le p}(a_jb_jc_j)^E=M^3$.

Our (4,1,4)⊕(1,9,1) case is especially favorable for the ASI, not only because it uniquely minimizes E over all cases of Schönhage's lemma 6.1, and is maximally-simple, but also because the "symmetrization step" in the proof of the ASI may be skipped because (4,1,4) and (1,9,1) both are palindromes.

We now shall figure out the breakeven N for that Schönhage 1981 matrix multiplication method while at the same time hopefully making it clear why and how the ASI works. Schönhage's first point is that $Rk_{2P}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes P}]\le 17^P$. The Pth tensor power of (4,1,4)⊕(1,9,1) is seen (due to expanding it via the "binomial theorem") to contain binomial(P,a)=binomial(P,b) copies of $(4^a,9^b,4^a)$ for each pair (a,b) of non-negative integers with a+b=P. Here binomial(A,B)≡A!/[B!(A-B)!]. If we further demand $N=4^a=9^b$ then $(4^a,9^b,4^a)$=(N,N,N). Actually you cannot satisfy this demand exactly with *integer* a,b, but if P is large enough to ignore round-to-integer error then we find a=(log3/log6)P≈0.613147P and b=(log2/log6)P≈0.386853P. We therefore can *approximately* multiply two N×N matrices with $N=4^{(\log3/\log6)P}\approx2.33965^P$, hence with $N^3\approx12.8072^P$, in $17^P$/binomial(P,a) muls *per problem*, with approximation degree=2P. Hence we can *exactly* multiply them in $(2P+1)17^P$/binomial(P,P·log3/log6) muls *per problem* where note we are solving binomial(P,P·log3/log6) independent problems of this type simultaneously. Solving for least integer P>1 such that #muls≤$N^3$ we find P=13. Therefore breakeven $N_{obv}=2.33965^{13}\approx62945.06$ would work – *if* we did not need to worry about round-to-integer effects on matrix sizes.

It seems to me those effects can be handled by filling a fraction≤O(1/P) of the rows or columns of some of our matrices with 0s.

More precise than that general idea, though, is to consider specific numbers and the specific roundings-to-integers arising from them. Specifically $Rk_{26}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes13}]\le17^{13}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes13}$ contains binomial(13,5)=binomial(13,8)=1287 copies of $(4^8,9^5,4^8)$=(65536,59049,65536). Therefore, 59049×59049 matrices can be multiplied in $17^{13}27/1287\approx2.07788\times10^{14}$ bilinear multiplications per copy, which slightly exceeds the obvious method's $59049^3\approx2.05891\times10^{14}$. So the choice P=13 does not quite work to beat Obvious. Nor does P=14.

With **P=15**, we have $Rk_{30}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes15}]\le17^{15}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes15}$ contains binomial(15,6)=binomial(15,9)=5005 copies of $(4^9,9^6,4^9)$=(262144,531441,262144). Therefore, N×N matrices with $N=262144=2^{18}$ can be multiplied in $17^{15}31/5005\approx1.77293\times10^{16}$ bilinear multiplications per copy, which is less than the obvious method's $262144^3=2^{54}\approx1.80144\times10^{16}$. Hence the breakeven $N_{obv}=262144=2^{18}$.

If we instead solve for the least integer P>1 such that #muls≤$N^S$ we would find, ignoring integer roundoff effects, P≈27. But the roundoffs again are unfavorable. Specifically:

**P=27:** $Rk_{54}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes27}]\le17^{27}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes27}$ contains binomial(27,16)=binomial(27,11)=13037895 copies of $(4^{16},9^{11},4^{16})$=(4294967296, 31381059609, 4294967296). Therefore, N×N matrices with $N=4^{16}=2^{32}$ can be multiplied in $17^{27}55/13037895\approx7.035\times10^{27}$ bilinear multiplications per copy, which is less than the obvious method's $2^{96}\approx79.228\times10^{27}$ but not as good as Strassen's $7^{32}\approx1.104\times10^{27}$.

**P=28:** $Rk_{56}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes28}]\le17^{28}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes28}$ contains binomial(28,17)=binomial(28,11)=21474180 copies of $(4^{17},9^{11},4^{17})$=(17179869184, 31381059609, 17179869184). Therefore, N×N matrices with $N=4^{17}=2^{34}$ can be multiplied in $17^{28}57/21474180\approx7.525\times10^{28}$ bilinear multiplications per copy, which is less than the obvious method's $2^{102}\approx507.0602\times10^{28}$ but not as good as Strassen's $7^{34}\approx5.412\times10^{28}$.

**P=29:** $Rk_{58}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes29}]\le17^{29}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes29}$ contains binomial(29,18)=binomial(29,11)=34597290 copies of $(4^{18},9^{11},4^{18})$=(68719476736, 31381059609, 68719476736). Therefore, $9^{11}\times9^{11}$ matrices can be multiplied in $17^{29}59/34597290\approx8.219\times10^{29}$ bilinear multiplications per copy, which is much less than the obvious method's $3^{66}\approx309.032\times10^{29}$ but not as good as Strassen's #muls≤$7^{35}\approx3.788\times10^{29}$.

**P=30:** $Rk_{60}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes30}]\le17^{30}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes30}$ contains binomial(30,18)=binomial(30,12)=86493225 copies of $(4^{18},9^{12},4^{18})$=(68719476736, 282429536481, 68719476736). Therefore, N×N matrices with $N=4^{18}=2^{36}$ can be multiplied in $17^{30}61/86493225\approx5.7785\times10^{30}$ bilinear multiplications per copy, which is much less than the obvious method's $2^{108}\approx324.519\times10^{30}$ but not as good as Strassen's $7^{36}\approx2.652\times10^{30}$.

Finally with **P=31**, we get $Rk_{62}[\{(4,1,4)\oplus(1,9,1)\}^{\otimes31}]\le17^{31}$, and $\{(4,1,4)\oplus(1,9,1)\}^{\otimes31}$ contains binomial(31,19)=binomial(31,12)=141120525 copies of $(4^{19},9^{12},4^{19})$=(274877906944, 282429536481, 274877906944). Therefore, N×N matrices with $N=4^{19}=2^{38}$ can be multiplied in $17^{31}63/141120525\approx6.218\times10^{31}$ bilinear multiplications per copy, which is much less than the obvious method's $2^{114}\approx2076.919\times10^{31}$ and (finally!) also beats Strassen's $7^{38}\approx12.9935\times10^{31}$, indeed by a factor>2. So the breakeven $N_{str}=2^{38}\approx2.749\times10^{11}$.

It is quite interesting that this Schönhage ASI method (shown yellow in the table) actually does much *better* then any of the plain APA methods tabulated (shown pink) reckoned by either E, $N_{obv}$, or $N_{str}$. It would appear to obsolete every known plain-APA method. (Admittedly Schönhage ASI solves *many*, not just one, matrix-multiplication problem; and it wastefully computes up to about P times as many output quantities as we want, simply wasting both compute time and memory before we discarded them; but even taking both those complaints into account all known plain-APA's still seem obsoleted reckoned by either E or $N_{str}$. Furthermore, *if* Strassen's additivity conjecture is correct, then for each N there must exist a *single copy* N×N matrix-multiplication algorithm with the same or smaller mul-count as our ASI-method's mul-count *per copy*.)

## Strassen 1987's "laser method": E≤ln(54)/ln(5)≈2.4784951415313494

Strassen's idea is quite similar in essence to Schönhage's ASI. (Incidentally, this whole construction of Strassen's is helpfully re-explained by Coppersmith & Winograd 1990 in their §3. Here I am going to provide a simpler related method I call "simplified laser" which yields the same exponents E.) Both consider high ⊗-powers of some nice APA method T, then find inside the resulting messes, many copies of efficient APA formulas for multiplying large square matrices. (There also is a lot of other stuff there too – which we simply discard, ignore, and waste!) The difference is that Strassen-laser, unlike Schönhage-ASI, does not demand that T be a ⊕ of rectangular matrix product tasks. Strassen allows a wider class of bilinear tasks T that can bear little to no resemblance to any matrix product task. Nevertheless at the end of the game, Strassen still finds inside the resulting messes, many copies of efficient APA formulas for multiplying large matrices, albeit these copies can occur somewhat "**encrypted**," e.g. with their entries permuted. (Decryptions do not increase the bilinear-mul count.)

Strassen stated an $APA_1$ formula for a certain family (parameterized by an integer q) of tasks T with $Rk_1[T]\le q+1$. Hence $Rk_p[T^{\otimes p}]\le(q+1)^p$ and $Rk_{3p}[T^{\otimes p}\otimes T'^{\otimes p}\otimes T''^{\otimes p}]\le(q+1)^{3p}$. He then argued that $T^{\otimes p}\otimes T'^{\otimes p}\otimes T''^{\otimes p}$ where T, T', and T'' are three symmetric altered versions of T, contains $2^{2p-2}3$ independent rectangular matrix multiplication tasks (a,b,c), with perhaps-differing (a,b,c)'s but in all cases obeying $abc=q^{3p}$. Then (essentially by the ASI), Strassen deduces E≤[3ln(q+1)-ln(4)]/ln(q). The best (E-minimizing) choice of q is q=5, yielding E≤ln(54)/ln(5)≈2.4784951415313494.

If q is prime (and 5 *is* prime), then the number of possible 3-tuples (a,b,c) of positive integers obeying $abc=q^{3p}$ equals 3(p+1)(3p+2)/2. Therefore, even if we restrict attention to the single most-popular 3-tuple (a,b,c), – just ignore and waste all the others – there must be at least $2^{2p}/[2(p+1)(3p+2)]$. This diminution is not enough to alter the asymptotic (when p→∞) value of E. The advantage of this "simplified laser method" is that we do not need to apply the ASI to many different (a,b,c)'s – we only have one type. Furthermore by the 3-rotation-symmetry of Strassen's T⊗T'⊗T'' and the classic "central limit theorem" governing binomial and trinomial coefficients, one may see that for large-enough p, the most popular (a,b,c) – or at least most popular up to arbitrarily small relative counting-error – should in fact have $a=b=c=q^p$. Therefore with this "simplified laser" we do not need the ASI since we are already where we want to be without it.

In summary (and converting from APA to exact, and assuming our p are large enough to make the "central limit theorem" valid enough for our purposes) my "simplified Strassen laser method" in this case employs $(q+1)^{3p}(3p+1)$ bilinear muls to multiply at least $2^{2p}/[2(p+1)(3p+2)]$ different pairs of N×N matrices with $N=q^p$, and the best choice of q (which for me must be prime) is q=5. *Per copy*, then, the mul-count for exact N×N matrix multiplication is $(q+1)^{3p}2^{1-2p}(p+1)(3p+2)(3p+1)$ which when q=5 simplifies to $6^{3p}2^{1-2p}(p+1)(3p+2)(3p+1)$. That beats Obvious, which uses $5^{3p}$, if p≥13, hence the breakeven $N_{obv}=5^{13}=1220703125\approx1.221\times10^9$ with $\log_2(N_{obv})\approx30.185$.

Laser's mul-count first goes below $M_{str1}(5^p)$ when p=23, when $N=5^{23}=11920928955078125\approx1.1921\times10^{16}$, whereupon $M_{str1}(N)\approx1.80966\times10^{45}$ and laser's mul-count≈$1.66976\times10^{45}$. Hence $N_{str}=5^{23}\approx5.960\times10^{16}$ with $\log_2(N_{str})\approx53.4043$.

My simplified Strassen laser method with E≈2.4785 again would appear to obsolete every known plain-APA method. However, I have (very roughly) calculated that it first beats Schönhage 1981's ASI method with E≈2.5480 for N somewhere between $N\approx10^{30}$ and $10^{60}$ with $100\le\log_2 N\le200$.

## Interlude about Salem-Spencer function

If X≥0 is an integer, **SalemSpencer(X)** denotes the greatest cardinality of a subset of {1,2,3,...,X} containing no 3-term arithmetic progression. (Or "a subset of {J,J+1,J+2,...,J-1+X}" for any particular J, e.g. starting from J=0 is often more convenient, and J does not even need to be an integer.) For example SalemSpencer(14)=8 because of the unique {1,2,4,5,10,11,13,14}. These sets have also been called "nonaveraging sets" since no element is the average of any other two: if a,b,c∈S then a+c=2b ⇒ a=b=c.

**The table:** The primary purpose of this entire section is to compile a big table – by far the best available – of bounds on SalemSpencer(X), for X up to about $10^{50}$. All tabulated SalemSpencer "values">1024 really are merely *lower bounds* (since I got tired of writing "≥"). Bounds stated with decimal points are inexact, since computed by Monte Carlo, but have ≤0.001 relative standard statistical errors. "×N": indicates a "product" using ModSS(X)≥N. When SalemSpencer(X)≥B we also tabulate the "exponent" e=log(B)/log(X) obeying $X^e=B$. Although Behrend's constructions show e→1 when X→∞, the greatest e achieved in the table for X>100 is only e≈0.769 arising from $X\approx2.54254\times10^{50}$ and $B\approx6.281\times10^{38}$ via the Tbh(34,31,351) Triangular Behrend construction.
**Key:** b3,b5,b7=Base 3,5,7 bounds by Szekeres, Rusza, and me. EX=exhaustive searches for X≤211 by Fausto A.C. Cariboni. GGK=incomplete "branch & bound" computer searches by Gasarch, Glenn, Kruskal 2008. GT & JW=computer searches by Gavin Theobald & Jaroslaw Wroblewski. Tbh(D,R,V): My "triangular Behrend" construction (described below). WY=J.Wroblewski & Fumitaka Yura computer search.

There is an **O(XlogX)-op algorithm** which, given a subset of {0,1,2,...,X-1}, decides whether it is a nonaveraging set (and if not, finds a counterexample, indeed finds *all* midpoints b of 3-term-arithmetic progressions):

1. Sort the set into increasing order in O(XlogX) steps.
2. Let $a_j=1$ if j is in the set, otherwise $a_j=0$.
3. Let P(u) be the polynomial $\sum_{0\le j<X} u^{(a_j)}$.
4. Compute $P(u)^2$, which takes O(XlogX) operations using fast convolution algorithms based on the FFT.
5. If all coefficients of $P(u)^2$ are <3 then output "set is nonaveraging" and stop.
6. If any coefficient of $u^{2b}$ is ≥3 then output "set contains 3-term arithmetic progression a,b,c" and for any particular such b we can search for suitable a (going leftward from b) and c (going rightward from b the same distance) in O(X) steps.

| X | SalemSpencer(X) | ⌊1000e⌋ | Reason |
|---|---|---|---|
| 1 | 1 | | |
| 2-3 | 2 | 1000 | b3(1),EX |
| 4 | 3 | 792 | EX |
| 5-8 | 4 | 861 | b3(2),EX |
| 11-12 | 6 | 747 | EX |
| 14-19 | 8 | 787 | b3(3),EX |
| 20-23 | 9 | 733 | EX |
| 26-29 | 11 | 735 | EX |
| 32-35 | 13 | 740 | EX |
| 41-50 | 16 | 746 | b3(4),EX |
| 63-70 | 20 | 723 | EX |
| 100-103 | 27 | 715 | EX |
| 122-136 | 32 | 721 | b3(5),EX |
| 174-193 | 40 | 715 | EX |
| 209-211 | 43 | 704 | EX |
| 222 | 44-52 | 700 | 22×2,GGK |
| 227 | 45-55 | 701 | GGK |
| 233 | 46-56 | 702 | GT,GGK |
| 245 | 48-58 | 703 | GT,GGK |
| 256 | ≥49 | 701 | GT |
| 272 | ≥53 | 708 | GT |
| 332 | ≥61 | 708 | GT |
| 365 | ≥64 | 704 | b3(6) |
| 518 | ≥80 | 701 | 8×10 |
| 768 | ≥102 | 696 | GT |
| 809 | ≥108 | 699 | GT |
| 864 | ≥114 | 700 | GT |
| 916 | ≥119 | 700 | GT |
| 1023 | ≥126 | 697 | GT |
| 1092 | $128 = 2^7$ | 693 | GT |
| 1241 | ≥151 | 704 | GT |
| 1375 | ≥160 | 702 | 4×40 |
| 1410 | ≥161 | 700 | GT |
| 1881 | ≥172 | 682 | 43×4 |
| 1998 | ≥180 | 683 | 18×10 |
| 2548 | ≥224 | 689 | 8×28 |
| 2828 | ≥248 | 693 | 8×31 |
| 3180 | $\geq 256 = 2^8$ | 687 | GT |
| 3850 | ≥320 | 698 | 8×40 |
| 5500 | ≥360 | 683 | 9×40 |
| 7150 | ≥440 | 685 | 11×40 |
| 8800 | $\geq 520 > 2^9$ | 688 | 13×40 |
| 9900 | ≥560 | 687 | 14×40 |
| 11275 | ≥640 | 692 | 16×40 |
| 14025 | ≥680 | 683 | 17×40 |
| 14850 | ≥720 | 684 | 18×40 |
| 15950 | ≥760 | 685 | 19×40 |
| 17325 | ≥800 | 684 | 20×40 |
| 26125 | $1040 > 2^{10}$ | 683 | 26×40 |
| 33550 | 1280 | 686 | 32×40 |
| 46475 | 1560 | 684 | 39×40 |

| X | SalemSpencer(X) | ⌊1000e⌋ | Reason |
|---|---|---|---|
| 53350 | 1640 | 680 | 41×40 |
| 57475 | 1720 | 679 | 43×40 |
| 78375 | $2080 > 2^{11}$ | 677 | 1040×2 |
| 100375 | 2560 | 681 | b3(6)×40 |
| 235125 | $4160 > 2^{12}$ | 673 | $1040\times 2^2$ |
| 237600 | 4560 | 680 | GT×40 |
| 300300 | 5120 | 677 | GT×40 |
| 387750 | 6440 | 681 | GT×40 |
| 705375 | $8320 > 2^{13}$ | 670 | 4160×2 |
| 1058750 | 12800 | 681 | $8\times 40^2$ |
| 1966250 | 17600 | 674 | $11\times 40^2$ |
| 2116125 | $16640 > 2^{14}$ | 667 | ×2 |
| 2420000 | 20800 | 676 | $13\times 40^2$ |
| 4764375 | 32000 | 674 | $20\times 40^2$ |
| 6348375 | $33280 > 2^{15}$ | 664 | ×2 |
| 7184375 | 41600 | 673 | $26\times 40^2$ |
| 12780625 | 62400 | 674 | $39\times 40^2$ |
| 15805625 | $68800 > 2^{16}$ | 671 | $43\times 40^2$ |
| 20570000 | 84800 | 673 | $53\times 40^2$ |
| 25107500 | 97600 | 674 | $61\times 40^2$ |
| 47416875 | $137600 > 2^{17}$ | 669 | ×2 |
| 106631250 | 257600 | 674 | $161\times 40^2$ |
| 141232023 | 258048 | 664 | b7(5) |
| 142250625 | $275200 > 2^{18}$ | 667 | ×2 |
| 291156250 | 512000 | 674 | $8\times 40^3$ |
| 423696069 | 516096 | 662 | b7(5)×2 |
| 426751875 | $550400 > 2^{19}$ | 665 | ×2 |
| 540718750 | 704000 | 669 | $11\times 40^3$ |
| 665500000 | 832000 | 670 | $13\times 40^3$ |
| 852671875 | 1024000 | 672 | $b3(4)\times 40^3$ |
| 1310203125 | $1280000 > 2^{20}$ | 669 | $20\times 40^3$ |
| 2079687500 | 1728000 | 669 | $27\times 40^3$ |
| 2537218750 | 2048000 | 671 | $b3(5)\times 40^3$ |
| 3536728278 | $2.107e6 > 2^{21}$ | 662 | Tbh(9,11,12)×2 |
| 4346546875 | 2752000 | 668 | $43\times 40^3$ |
| 6920604385 | 3784704 | 668 | b7(6)=Tbh(12,7,6) |
| 10610184834 | $4.215e6 > 2^{22}$ | 660 | Tbh(9,11,12)×4 |
| 12968325779 | 5.960e6 | 669 | Tbh(10,11,13) |
| 19219943492 | 7.890e6 | 670 | Tbh(9,15,23) |
| 28528868646 | 8.000e6 | 660 | Tbh(10,7,5)×31 |
| 36579093772 | $9.461e6 > 2^{23}$ | 660 | Tbh(11,7,5)×10 |
| 38838806325 | 1.032e7 | 662 | Tbh(10,7,5)×40 |
| 48444465988 | 1.406e7 | 668 | Tbh(13,7,7) |
| 62285439465 | 15138816 | 665 | b7(6)×4 |
| 80067968750 | 20480000 | 670 | $8\times 40^4$ |
| 116714932011 | 2.384e7 | 666 | Tbh(10,11,13)×4 |
| 142655126682 | $3.417e7 > 2^{25}$ | 675 | Tbh(11,11,15) |

| X | SalemSpencer(X) | ⌊1000e⌋ | Reason |
|---|---|---|---|
| 288321933617 | 5.875e7 | 677 | Tbh(10,15,26) |
| 427965380046 | 6.835e7 > $2^{26}$ | 673 | Tbh(11,11,15)×2 |
| 711137909204 | 7.890e7 | 666 | Tbh(9,15,23)×10 |
| 864965800851 | 1.175e8 | 676 | Tbh(10,15,26)×2 |
| 1007990560869 | 1.216e8 | 673 | Tbh(10,17,43) |
| 1283896140138 | 1.367e8 > $2^{27}$ | 671 | Tbh(11,11,15)×4 |
| 1569209936615 | 1.973e8 | 680 | Tbh(12,11,16) |
| 2373778833372 | 2.109e8 | 672 | Tbh(15,7,7) |
| 2594897402553 | 2.350e8 | 674 | Tbh(10,15,26)×4 |
| 3851688420414 | 2.734e8 > $2^{28}$ | 670 | Tbh(11,11,15)×8 |
| 4707629809845 | 3.947e8 | 678 | Tbh(12,11,16)×2 |
| 7784692207659 | 4.700e8 | 672 | Tbh(10,15,26)×8 |
| 8339897606041 | 6.270e8 > $2^{29}$ | 680 | Tbh(10,21,60) |
| 12974487012726 | 8.940e8 | 682 | Tbh(11,15,26)×2 |
| 17261312845878 | 1.140e9 > $2^{30}$ | 684 | Tbh(13,11,17) |
| 20712905678375 | 1.466e9 | 688 | Tbh(10,23,57) |
| 38923461038178 | 1.788e9 | 680 | Tbh(11,15,26)×4 |
| 47683145343751 | 2.414e9 > $2^{31}$ | 685 | Tbh(10,25,83) |
| 58244920356306 | 3.273e9 | 691 | Tbh(11,19,43) |
| 64872778301117 | 3.423e9 | 690 | Tbh(12,15,29) |
| 143049436031253 | 4.830e9 > $2^{32}$ | 684 | Tbh(10,25,83)×2 |
| 175138372880731 | 6.562e9 | 689 | Tbh(11,21,68) |
| 210353319238441 | 7.673e9 | 690 | Tbh(10,29,112) |
| 291310964560397 | 9.012e9 > $2^{33}$ | 688 | Tbh(12,17,52) |
| 409810293823897 | 1.457e10 | 695 | Tbh(10,31,102) |
| 631059957715323 | 1.535e10 | 688 | Tbh(10,29,112)×2 |
| 973096800297992 | 2.624e10 > $2^{34}$ | 695 | Tbh(13,15,32) |
| 1106655274513275 | 3.128e10 | 697 | Tbh(12,19,46) |
| 2088623609220854 | 3.826e10 > $2^{35}$ | 690 | Tbh(15,11,20) |
| 2919290400893976 | 5.249e10 | 693 | Tbh(13,15,32)×2 |
| 3319965823539825 | 6.257e10 | 695 | Tbh(12,19,46)×2 |
| 3677913026681293 | 6.917e10 > $2^{36}$ | 696 | Tbh(12,21,76) |
| 4952288038881425 | 7.784e10 | 693 | Tbh(13,17,57) |
| 8757871202681928 | 1.050e11 | 691 | Tbh(13,15,32)×4 |
| 9.9599e15 | 1.252e11 | 693 | Tbh(12,19,46)×4 |
| 1.10337e16 | 1.383e11 | 694 | Tbh(12,21,76)×2 |
| 1.45965e16 | 2.016e11 | 699 | Tbh(14,15,35) |
| 2.10265e16 | 3.004e11 | 703 | Tbh(13,19,52) |
| 2.98023e16 | 3.738e11 | 702 | Tbh(12,25,102) |
| 4.37894e16 | 4.031e11 | 697 | Tbh(14,15,35)×2 |
| 5.49023e16 | 4.392e11 | 695 | Tbh(11,29,124)×4 |
| 7.50473e16 | 8.964e11 | 708 | Tbh(12,27,93) |
| 2.18947e17 | 1.560e12 | 703 | Tbh(15,15,37) |
| 2.25142e17 | 1.793e12 | 706 | Tbh(12,27,93)×2 |
| 2.52018e17 | 2.219e12 | 709 | Tbh(13,23,75) |
| 3.99503e17 | 2.901e12 | 708 | Tbh(14,19,55) |
| 5.30722e17 | 3.161e12 | 705 | Tbh(12,29,132)×2 |

| X | SalemSpencer(X) | ⌊1000e⌋ | Reason |
|---|---|---|---|
| 7.45058e17 | 4.660e12 | 708 | Tbh(13,25,111) |
| 1.19851e18 | 5.803e12 | 706 | Tbh(14,19,55)×2 |
| 1.62196e18 | 7.720e12 | 707 | Tbh(14,21,87) |
| 3.2842e18 | 1.213e13 | 706 | Tbh(16,15,39) |
| 3.66263e19 | 1.044e14 | 716 | Tbh(13,31,133)×2 |
| 5.58794e19 | 1.169e14 | 712 | Tbh(14,25,118)×2 |
| 1.02183e20 | 1.642e14 | 710 | Tbh(15,21,95)×2 |
| 1.09879e20 | 2.087e14 | 714 | Tbh(13,31,133)×4 |
| 1.33318e20 | 2.977e14 | 719 | Tbh(15,23,86) |
| 1.48779e20 | 3.295e14 | 719 | Tbh(14,29,154) |
| 3.29637e20 | 4.174e14 | 712 | Tbh(13,31,133)×8 |
| 4.65661e20 | 7.333e14 | 719 | Tbh(15,25,127) |
| 7.15284e20 | 8.739e14 | 716 | Tbh(16,21,101) |
| 1.47716e21 | 2.202e15 | 724 | Tbh(15,27,117) |
| 3.06631e21 | 3.461e15 | 723 | Tbh(16,23,92) |
| 4.31459e21 | 4.776e15 | 724 | Tbh(15,29,164) |
| 9.19892e21 | 6.922e15 | 721 | Tbh(16,23,92)×2 |
| 1.17326e22 | 1.244e16 | 729 | Tbh(15,31,155) |
| 3.51979e22 | 2.489e16 | 727 | Tbh(15,31,155)×2 |
| 3.98832e22 | 2.985e16 | 728 | Tbh(16,27,126) |
| 7.0525e22 | 4.033e16 | 726 | Tbh(17,23,98) |
| 1.25123e23 | 6.941e16 | 729 | Tbh(16,29,177) |
| 2.11575e23 | 8.065e16 | 724 | Tbh(17,23,98)×2 |
| 9.8921e23 | 2.497e17 | 725 | Tbh(19,19,75) |
| 1.07685e24 | 4.053e17 | 732 | Tbh(17,27,133) |
| 1.62208e24 | 4.706e17 | 729 | Tbh(18,23,103) |
| 3.62857e24 | 1.011e18 | 733 | Tbh(17,29,188) |
| 7.27596e24 | 1.472e18 | 730 | Tbh(18,25,154) |
| 1.08857e25 | 2.021e18 | 731 | Tbh(17,29,188)×2 |
| 1.12751e25 | 2.996e18 | 737 | Tbh(17,31,176) |
| 2.90749e25 | 5.521e18 | 736 | Tbh(18,27,142) |
| 3.38252e25 | 5.992e18 | 735 | Tbh(17,31,176)×2 |
| 1.05229e26 | 1.474e19 | 736 | Tbh(18,29,199) |
| 1.81899e26 | 1.863e19 | 733 | Tbh(19,25,162) |
| 2.61674e26 | 2.208e19 | 732 | Tbh(18,27,142)×4 |
| 3.04427e26 | 2.397e19 | 731 | Tbh(17,31,176)×8 |
| 3.49527e26 | 4.661e19 | 740 | Tbh(18,31,186) |
| 7.85021e26 | 7.523e19 | 739 | Tbh(19,27,150) |
| 1.04858e27 | 9.323e19 | 739 | Tbh(18,31,186)×2 |
| 3.05163e27 | 2.153e20 | 739 | Tbh(19,29,211) |
| 4.54747e27 | 2.362e20 | 736 | Tbh(20,25,170) |
| 7.06519e27 | 3.009e20 | 735 | Tbh(19,27,150)×4 |
| 1.08353e28 | 7.261e20 | 744 | Tbh(19,31,196) |
| 2.11956e28 | 1.027e21 | 741 | Tbh(20,27,158) |
| 3.2506e28 | 1.452e21 | 742 | Tbh(19,31,196)×2 |
| 6.35867e28 | 2.054e21 | 739 | Tbh(20,27,158)×2 |
| 8.84973e28 | 3.149e21 | 742 | Tbh(20,29,222) |
| 1.9076e29 | 4.108e21 | 738 | Tbh(20,27,158)×4 |
| 3.35895e29 | 1.133e22 | 746 | Tbh(20,31,207) |

| X | SalemSpencer(X) | ⌊1000e⌋ | Reason | X | SalemSpencer(X) | ⌊1000e⌋ | Reason |
|---|---|---|---|---|---|---|---|
| 5.72281e29 | 1.404e22 | 744 | Tbh(21,27,166) | 2.68297e39 | 6.680e29 | 756 | Tbh(26,31,270)×4 |
| 1.00769e30 | 2.266e22 | 745 | Tbh(20,31,207)×2 | 4.5797e39 | 9.282e29 | 755 | Tbh(27,29,300)×2 |
| 1.71684e30 | 2.808e22 | 742 | Tbh(21,27,166)×2 | 5.98626e39 | 1.284e30 | 756 | Tbh(28,27,222) |
| 2.56642e30 | 4.611e22 | 745 | Tbh(21,29,232) | 8.04892e39 | 1.336e30 | 754 | Tbh(26,31,270)×8 |
| 5.15053e30 | 5.616e22 | 740 | Tbh(21,27,166)×4 | 9.24136e39 | 2.623e30 | 761 | Tbh(27,31,281) |
| 7.69926e30 | 9.222e22 | 743 | Tbh(21,29,232)×2 | 2.77241e40 | 5.245e30 | 759 | Tbh(27,31,281)×2 |
| 1.04128e31 | 1.770e23 | 749 | Tbh(21,31,217) | 4.42705e40 | 6.837e30 | 758 | Tbh(28,29,311) |
| 1.54516e31 | 1.920e23 | 746 | Tbh(22,27,174) | 8.31722e40 | 1.049e31 | 758 | Tbh(27,31,281)×4 |
| 2.30978e31 | 1.844e23 | 741 | Tbh(21,29,232)×4 | 1.61629e41 | 1.766e31 | 758 | Tbh(29,27,230) |
| 3.12383e31 | 3.540e23 | 747 | Tbh(21,31,217)×2 | 2.49517e41 | 2.098e31 | 756 | Tbh(27,31,281)×8 |
| 4.63547e31 | 3.841e23 | 744 | Tbh(22,27,174)×2 | 2.86482e41 | 4.122e31 | 762 | Tbh(28,31,291) |
| 7.44262e31 | 6.761e23 | 747 | Tbh(22,29,245) | 8.59446e41 | 8.243e31 | 761 | Tbh(28,31,291)×2 |
| 9.37148e31 | 7.080e23 | 745 | Tbh(21,31,217)×4 | 1.28384e42 | 1.008e32 | 760 | Tbh(29,29,322) |
| 1.39064e32 | 7.682e23 | 743 | Tbh(22,27,174)×4 | 2.57834e42 | 1.649e32 | 759 | Tbh(28,31,291)×4 |
| 2.23279e32 | 1.352e24 | 745 | Tbh(22,29,245)×2 | 4.36398e42 | 2.432e32 | 759 | Tbh(30,27,237) |
| 3.22795e32 | 2.767e24 | 751 | Tbh(22,31,228) | 8.88094e42 | 6.481e32 | 763 | Tbh(29,31,302) |
| 4.17193e32 | 2.631e24 | 748 | Tbh(23,27,182) | 2.66428e43 | 1.296e33 | 762 | Tbh(29,31,302)×2 |
| 6.69836e32 | 2.704e24 | 744 | Tbh(22,29,245)×4 | 3.72314e43 | 1.487e33 | 761 | Tbh(30,29,333) |
| 9.68386e32 | 5.534e24 | 750 | Tbh(22,31,228)×2 | 7.99285e43 | 2.592e33 | 761 | Tbh(29,31,302)×4 |
| 1.77636e33 | 6.168e24 | 745 | Tbh(24,25,205) | 1.17828e44 | 3.350e33 | 760 | Tbh(31,27,246) |
| 2.15836e33 | 9.923e24 | 749 | Tbh(23,29,255) | 2.75309e44 | 1.020e34 | 765 | Tbh(30,31,312) |
| 2.90516e33 | 1.107e25 | 748 | Tbh(22,31,228)×4 | 8.25928e44 | 2.039e34 | 763 | Tbh(30,31,312)×2 |
| 5.32907e33 | 1.234e25 | 743 | Tbh(24,25,205)×2 | 1.07971e45 | 2.194e34 | 762 | Tbh(31,29,345) |
| 6.47508e33 | 1.985e25 | 748 | Tbh(23,29,255)×2 | 2.47778e45 | 4.078e34 | 762 | Tbh(30,31,312)×4 |
| 1.00067e34 | 4.331e25 | 754 | Tbh(23,31,238) | 3.18134e45 | 4.616e34 | 761 | Tbh(32,27,254) |
| 2.74376e34 | 4.704e25 | 745 | Tbh(22,31,228)×17 | 7.43335e45 | 8.157e34 | 761 | Tbh(30,31,312)×8 |
| 3.002e34 | 8.663e25 | 752 | Tbh(23,31,238)×2 | 8.53459e45 | 1.605e35 | 766 | Tbh(31,31,322) |
| 6.25925e34 | 1.458e26 | 751 | Tbh(24,29,266) | 2.56038e46 | 3.210e35 | 765 | Tbh(31,31,322)×2 |
| 9.00599e34 | 1.733e26 | 750 | Tbh(23,31,238)×4 | 3.13116e46 | 3.240e35 | 763 | Tbh(32,29,356) |
| 1.87777e35 | 2.915e26 | 750 | Tbh(24,29,266)×2 | 7.68113e46 | 6.421e35 | 763 | Tbh(31,31,322)×4 |
| 2.7018e35 | 3.465e26 | 749 | Tbh(23,31,238)×8 | 8.58963e46 | 6.367e35 | 762 | Tbh(33,27,262) |
| 3.10206e35 | 6.786e26 | 755 | Tbh(24,31,249) | 9.39349e46 | 6.480e35 | 762 | Tbh(32,29,356)×2 |
| 9.30619e35 | 1.357e27 | 754 | Tbh(24,31,249)×2 | 1.69407e47 | 7.157e35 | 759 | Tbh(34,25,291) |
| 1.81518e36 | 2.143e27 | 753 | Tbh(25,29,277) | 2.64572e47 | 2.528e36 | 767 | Tbh(32,31,332) |
| 2.79186e36 | 2.715e27 | 752 | Tbh(24,31,249)×4 | 7.93717e47 | 5.056e36 | 766 | Tbh(32,31,332)×2 |
| 5.44554e36 | 4.285e27 | 752 | Tbh(25,29,277)×2 | 1.98002e48 | 5.137e36 | 760 | Tbh(31,31,322)×32 |
| 8.2116e36 | 6.796e27 | 753 | Tbh(26,27,206) | 2.38115e48 | 1.011e37 | 764 | Tbh(32,31,332)×4 |
| 9.6164e36 | 1.064e28 | 757 | Tbh(25,31,259) | 8.20174e48 | 3.984e37 | 768 | Tbh(33,31,343) |
| 2.88492e37 | 2.128e28 | 756 | Tbh(25,31,259)×2 | 2.24886e49 | 4.297e37 | 762 | Tbh(32,31,332)×17 |
| 5.26403e37 | 3.152e28 | 755 | Tbh(26,29,289) | 2.46052e49 | 7.969e37 | 767 | Tbh(33,31,343)×2 |
| 8.65476e37 | 4.256e28 | 754 | Tbh(25,31,259)×4 | 6.26184e49 | 1.212e38 | 764 | Tbh(35,27,278) |
| 1.57921e38 | 6.304e28 | 753 | Tbh(26,29,289)×2 | 7.38156e49 | 1.594e38 | 766 | Tbh(33,31,343)×4 |
| 2.21713e38 | 9.338e28 | 755 | Tbh(27,27,214) | 2.21447e50 | 3.188e38 | 764 | Tbh(33,31,343)×8 |
| 2.98108e38 | 1.670e29 | 759 | Tbh(26,31,270) | 2.54254e50 | 6.281e38 | 769 | Tbh(34,31,354) |
| 6.6514e38 | 1.868e29 | 753 | Tbh(27,27,214)×2 | 7.62762e50 | 1.256e39 | 768 | Tbh(34,31,354)×2 |
| 8.94325e38 | 3.340e29 | 757 | Tbh(26,31,270)×2 | 1.69070e51 | 1.673e39 | 765 | Tbh(36,27,286) |
| 1.52657e39 | 4.641e29 | 757 | Tbh(27,29,300) | 2.28828e51 | 2.513e39 | 767 | Tbh(34,31,354)×4 |

Due to exhaustive computer searches, SalemSpencer(X) is known exactly for each X≤211. Unfortunately, it seems entirely possible that nobody will ever know any exact value of SalemSpencer(X) for any X≥2000. For all large-enough X, there exist positive constants $c_1$, $c_2$, $c_3$, $c_4$, such that

$$c_1 X \exp(-c_2 [\lg X]^{1/2}) (\lg X)^{1/4} < \text{SalemSpencer}(X) < c_3 X \exp(-c_4 [\lg X]^{1/9})$$

Here $\lg X \equiv \log_2 X$, the lower bound is Elkin 2011's improvement of Behrend 1946 which both have $c_2 = 8^{1/2}\ln 2 < 1.9605163$, and the upper bound was proved by Kelley & Meka as refined by Bloom & Sisask in 2023 with $c_3=1$ for all large-enough X. Miscellaneous bounds:

- **Monotonicity & Subadditivity:** SalemSpencer(X)≤SalemSpencer(X+Y)≤SalemSpencer(X)+SalemSpencer(Y).
- **Cubes:** The sequence of nonnegative cubes 0,1,8,27,64,125,... contains no three elements in arithmetic progression (theorem by L.Euler 1770 and by A-M.Legendre 1823 or before, see Dickson vol2 p.572-573). Indeed Darmon & Merel 1997 showed for each K≥3 that there are no three nonnegative Kth-powers in arithmetic progression, proving a conjecture of Denes 1952, who had already proven it for K∈{3,4,5,...,30} in his Satz 9. Hence SalemSpencer(X)≥$\lfloor 1+X^{\mathbf{1/3}} \rfloor$. But that does **not** work for squares (K=2) due to (1,25,49), nor for triangle numbers due to (6,21,36) – although numerous authors starting with Euler in 1780 all proved or repeated a 1640 claim by Fermat that no *four* squares form an arithmetic progression (Dickson vol.2 p.440 and near bottom of p.635 lists citations, few or none of which actually contain proofs; Itard 1963 allegedly proved it by "Fermat's method of infinite descent"

on his p.112-113; the webpage by Brown gives another; Conrad uses an elliptic curve to prove in his "theorem 3.4" the somewhat stronger claim that no four *rational* squares form an arithmetic progression; also no 4 triangular numbers form an arithmetic progression for the same reason).

- **Certain squares or triangular numbers:** Nevertheless, the squares $n^2$ arising only from n that are products of distinct primes of form 4j+3 (examples: n=19 and n=3·7·23) is a nonaveraging set. So are the triangular numbers (n+1)n/2 arising only from n with 2n+1 equalling such a product. Consequently, $SalemSpencer(X)\ge c(X/\log X)^{1/2}$ for some positive constant c. [Proof sketch: L.Pisano ("Fibonacci") in 1225 proved a theorem completely characterizing the triples (a,b,c) of nonnegative integers such that $a^2+c^2=2b^2$, namely: $a=(k/2)(x^2+2xy-y^2)$, $b=(k/2)(x^2+y^2)$, $c=(k/2)(y^2+2xy-x^2)$ for any 0<x<y and k>0 such that a,b,c all are integers, Then in view of the famous two-squares theorem (originally stated by A.Girard in 1632) that a number z is representable as a sum of two squares $z=x^2+y^2$ if and only if all the primes of form 4j+3 in its prime factorization occur only raised to *even* powers, we see that if we restrict attention only to the $n^2$ with n being products of distinct primes all of form 4j+3, then the middle elements $b^2$ of 3-term arithmetic progressions of squares, can never occur. Finally, the fact that the numbers up to X≥3 that are products of distinct primes all of form 4j+3, is known to have cardinality asymptotic to a positive constant times $(X/\log X)^{1/2}$ by arguments related to the "Landau-Ramanujan constant."]
- **Randomizing argument:** Permute the elements of {1,2,3,...,X} into random order than add them one at a time to the set (when permitted). The expected cardinality of the final set, which lower-bounds SalemSpencer(X), will be $\ge 2X^{\mathbf{1/2}}/3$.
- **Explicit square-based construction:** Let P be a prime. Consider the set of 2-digit numbers AB written in radix R=2P+1 with each digit in {0,1,2,...,P-1} [causing addition of two such numbers to be carryless] and $B=A^2-1$ mod P. Then this set is nonaveraging and shows SalemSpencer(2[P-1]P+1)≥P. Apparently this construction never is optimal. Hence for an infinite set of X we have $SalemSpencer(X)\ge[(2X-1)^{1/2}+1]/2$ and in view of known results about gaps between consecutive primes $SalemSpencer(X)\ge(X/2)^{1/2}[1-o(1)]$ so that $SalemSpencer(X)>0.7071X^{1/2}$ for all large-enough X.
- **G.Szekeres' Base-3 method:** Consider the set of nonnegative numbers whose radix-3 representation contains no 2. For this integer subset a+c=2b is impossible. If we consider *only* the (≤k)-digit ternary numbers, this shows $SalemSpencer([3^k+1]/2)\ge 2^k$. That Szekeres bound is never optimal if k≥7, but nevertheless works well for $X<10^{10}$. Szekeres' bound also arises from "greed" instead of the above "randomizing argument," i.e. if you do *not* randomly pre-permute {0,1,2,...,X-1} but simply "greedily" add elements in increasing order when permitted, then you get precisely Szekeres' base-3 construction. For all large-enough X, Szekeres also shows $SalemSpencer(X)>X^{0.6309}$. since **0.6309**<log2/log3. Indeed for every X≥0 we have $SalemSpencer(X)>(½)(2X)^{\log 2/\log 3}$.
- **Products:** SalemSpencer(XY)≥ModSS(X)SalemSpencer(Y) where ModSS(Y) is the cardinality (preferably largest possible) of a nonaveraging subset of the integers modulo Y. [The nonaveraging set showing this is Xa+b where a is in the SalemSpencer(Y) set and b in the ModSS(X) set.] Also ModSS(XY)≥ModSS(X)ModSS(Y) and ModSS(X)≤SalemSpencer(X)≤ModSS(2X-1). These are very useful for "filling holes" in the table. For example, because ModSS(3)=2 [and hence $ModSS(3^k)\ge 2^k$ for each k≥0] we see that SalemSpencer(3Y)≥2SalemSpencer(Y), which enables forcing all "holes" in our table of SalemSpencer(X) lower bounds to be at most a factor 3 wide. More examples: ModSS(37)≥10 from {0,1,3,7,17,24,25,28,29,35} mod 37; ModSS(85)≥17 from {0,1,3,4,9,10,13,24,28,29,31,36,40,42,50,66,73} mod 85 (Fumitaka Yura); ModSS(182)≥28 from {0,5,6,9,22,23,29,31,32,34,43,48,50,51,60,61,75,84,85,92,101,103,104,106,112,129,130,135}; ModSS(202)≥31 from {0,9,10,17,19,22,23,40,42,43,47,56,59,60,66,68,79,81,87,88,91,100,104,105,107,125,128,130,137,138,147}; ModSS(232)≥32 from {0,3,8,15,17,21,33,36,40,46,50,53,62,68,76,81,82,87,95,101,110,113,117,123,127,130,142,146,148,155,160,163}; ModSS(275)≥40 from {0,7,8,10,17,22,23,28,40,42,43,45,53,54,59,60,87,88,93,94,102,104,105,107,119,124,125,130,137,139,140,147,177,183,199,209,213,223,239,245}. The last four all are by Gavin Theobald. The mod-232 one plus Szekeres shows, e.g, $SalemSpencer(232^k[3^j+1]/2)\ge 2^{5k+j}$ for all j,k≥0. The mod-275 one plus Szekeres similarly shows $SalemSpencer(275^k[3^j+1]/2)\ge 2^j 40^k$ for all j,k≥0, and hence $SalemSpencer(X)\ge X^{0.6567}$ for all large-enough X since **0.6567**<log(40)/log(275). Indeed for every X≥0 we have (after enough examination of small-X cases) $SalemSpencer(X)>X^{\log(40)/\log(275)}$.
- **New "Modular sum of two squares" construction:** Let P be prime with P mod 4=3 and hence unrepresentable as a sum of two squares [every multiple of P below $P^2$ also is unrepresentable]. Consider the set of 3-digit numbers ABC written in radix R=2P+1 with each digit in {0,1,2,...,P-1} [causing addition of two such numbers to be carryless] and $C=A^2+B^2-1$ mod P. Then this set is nonaveraging and shows $SalemSpencer([4P^2+2P-5]P+1)\ge P^2$ and hence for all large-enough X that $SalemSpencer(X)\ge(X/4)^{0.6666}$ since **0.6666**<2/3. Apparently this construction is never optimal.
- **I.Z.Ruzsa's Base-5 method:** Consider the set of nonnegative numbers written in radix 5 using digits 0,1,2 only with *exactly* a fixed count of 1s. Again for this integer subset a+c=2b is impossible. If we consider 3k-digit numbers using count=k, then the maximum permitted number is $222...222111...111_5$, i.e. (after evaluating the geometric sum) $[125^k 2-5^k-1]/4$, while the minimum is $000...000111...111_5=[5^k-1]/4$. Hence Rusza's construction shows $SalemSpencer([25^k-1]5^k/2+1)\ge 2^{2k}(3k)!/[(2k)!k!]$, which first outperforms Szekeres when k=6, showing SalemSpencer(1907348625001)≥76038144 while Szekeres only gives $67108864=2^{26}$. This 3k-digit Ruzsa bound apparently never is optimal. For all large-enough X this also shows $SalemSpencer(X)>X^{0.6826}$ since **0.6826**<log3/log5.
- **New Base-7 construction by me:** Consider the set of nonnegative numbers written in radix 7 using digits 0,1,2,3 only with *exactly* a fixed count of digits lying in the set {1,2}. For any such integer subset a+c=2b is impossible. If we consider 2k-digit numbers using count=k, this shows $SalemSpencer([(7^k 3-2)7^k+5]/6)\ge 2^{2k}(2k)!k!^{-2}$. This first exceeds Szekeres when k=3, showing SalemSpencer(58711)≥1280, versus 1024 from Szekeres, and k=4, showing SalemSpencer(2881601)≥17920, versus 16384 from Szekeres. For large X this also shows $SalemSpencer(X)>X^{0.7124}$ since **0.7124**<log4/log7.
- **L.Moser 1953's upper bound (obsoleted by below):** SalemSpencer(X)<min{2X/5+3, 4X/11+5}.
- **My new optimal-linear upper bound:** SalemSpencer(X)≤min{X, (2X+2)/3, (4X+16)/9, (8X+104)/27} with equality only when X=0, 1, 2, 5, 14, and 41. Proof: SalemSpencer(162)=36 by exhaustive search and since 162=6×27 and 36=6×6 we see that SalemSpencer(X)≤6X/27=2X/9 whenever X is a multiple of 162. For 0≤X≤211 the result holds by exhaustive verification. For X>211 it follows from subadditivity. Q.E.D.
I conjecture (16X+640)/81 can be adjoined as a fifth argument of the min, with X=122 added as a new equality-case. It even is somewhat plausible one can also adjoin (32X+1280)/243 and X=365, but this pattern definitely cannot be continued further. [Pattern? 0,2,16,104,640,1280 arise from $2^{n-1}(3^n-1)$, and 0,1,2,5,14,41,122,365 from $(3^n+1)/2$.]

**Jaroslaw Wroblewski's open question**: what is the minimum θ, such that for every B≥1, a cardinality=B nonaveraging subset of {1,2,3,..., $\lfloor B^\theta \rfloor$} exists? Because SalemSpencer(204)=42 we know that θ≥log(204)/log(42)>1.4228. Wroblewski conjectured θ≤3/2=1.50. I can prove θ≤log(275)/log(40)<1.522623 from ModSS(275)≥40 combined with known bounds for SalemSpencer(X) for small X. The data in my table is compatible with this conjecture: 1.42≤θ≤1.52. Molsen 1941 showed: Whenever N≥118 there are primes in (N,4N/3] congruent to each of 1, 5, 7, and 11 modulo 12. Consequently whenever N≥33 there are primes in (N,4N/3] congruent to 3 mod 4. This improved Breusch 1932's theorem that between every number N≥7 and 2N there is at least one prime number from each of the four progressions 3k+1, 3k+2, 4k+1 and 4k+3; for the cases 4k+1 and 4k+3 almost the same result was shown entirely using elementary methods by Erdös 1935 (redone more precisely by Moree 1993); those in turn improved "Bertrand's postulate" that for each N≥4 at least one prime p exists with N<p<2N-2. Anyhow, Molsen combined with my modular-sum-of-two-squares construction and explicit verifications for all small B shows that Wroblewski is correct if weakened by a factor of 10, that is: a cardinality=B nonaveraging subset of $\{1,2,3,..., \lfloor 10B^{3/2} \rfloor\}$ always exists.

**Behrend 1946's** lower bounds are ultimately superior to any of the above-listed miscellaneous constructions. I'll first explain his construction, then show how to improve it. It arises by choosing good parameters D,R,V in the following. Consider the set of D-digit numbers written in radix R=4J+1 but only permitting digits $x_k \in \{-J,1-J,...,J-1,J\}$. Here $x_k$, k=0,1,2,...,D-1, is the *k*th digit.) This prevents "carries" if two such numbers are added. Further demand that the $\sum_{0\le k<D}(x_k)^2=V$. We can call this set "Beh(D,R,V)." It cannot contain any 3-term arithmetic progression because the sphere of radius=$\sqrt{V}$ is a strictly-**convex** surface, whose intersection with a line therefore has at most *two* points.

The best choice of V is the one maximizing the cardinality #Beh(D,R,V) of Beh(D,R,V). We then have SalemSpencer(X)≥#Beh(D,R,V) where X=2Y+1 and Y is the maximum element of Beh(D,R,V). It is easy to determine Y using "rightward-traveling greed." E.g. if D=35, J=9 (hence R=37), and V=926, then the maximum element of Beh(D,R,V) is $99999999999531000000000000000000000_{37}$ in view of the fact that $926=9^2 11+5^2+3^2+1$. Note that, regardless of V, we always have $Y \le J\sum_{0\le k<D}R^k=(R^D-1)/4$. There are only $J^2D+1$ possible values V could take, namely $\{0,1,...,J^2D\}$. hence SalemSpencer$([R^D+1)2) \ge (2J+1)^D/(J^2D+1)$, which by choosing $D\approx(2\lg X)^{1/2}$ leads to Behrend's claim that $c_2=8^{1/2}\ln 2+o(1)<1.9605163$.

Rather stupidly, Behrend did not notice (in view of "Chebyshev's inequality" from probability theory, regarding the digits as being uniform-in-[-J,J] random integers) that only $O(J^2D^{1/2})$ of those candidate V – namely the ones within $\pm O(\sigma)$ away from Expectation(V)=(J+1)JD/3, where the standard deviation σ is asymptotic when $J\to\infty$ to $8(D/45)^{1/2}J^2$ – can allow substantial cardinalities; and they in total constitute at least a positive constant fraction of the summed cardinalities. Elkin 2011 understood this. Hence SalemSpencer$(R^D) \ge (2J+1)^D\kappa/(J^2D^{1/2}+1)$. for some positive constant κ. That does not improve Behrend's $c_2$, but does improve the bound-formula he gave at the start of his paper by a factor of order $(\lg X)^{1/4}$.

Further, if we employ the "central limit theorem" rather than merely Chebyshev's bound then we can determine the exact asymptotic (when D→∞, R→∞, V→∞) value of the constant κ, namely $\kappa=8^{-1}(2\pi/45)^{-1/2}\approx 0.3345233$.

So I recommend V≈(J+1)JD/3 and asymptotically $D\approx(2\lg X)^{1/2}$.

A second rather stupid decision by Behrend (unfortunately repeated by Elkin) was choosing the **squaring function** $f(x)=x^2$ in his demand that $\sum_{0\le k<D}f(x_k)=V$. A better choice would have been the *slowest-growing* concave-∪ integer→integer function, which is f(x)=(x+1)x/2 for x≥0 (*triangular* rather than square numbers). Asymptotically, this simple change improves Behrend's lower bounds on the SalemSpencer function by a factor of 2. More precisely:

**My new "triangular Behrend" construction:** Define the strictly-concave-∪ function $F_J(x)$ mapping integers to nonnegative integers as follows:

- If J is odd: $F_J(x)=(x-\lfloor J/2\rfloor)(x-\lceil J/2\rceil)/2$,
- If J is even: $F_J(x)=[2x-J][2x+2\operatorname{sign}(2x-J)-J]/8$.

Among all functions mapping integers x to nonnegative integers that are even-symmetric about x=J/2 and strictly-concave-∪, $F_J(x)$ is minimal for each x (this is easy to prove using induction on x→x+1). Consider the set Tbh(D,R,V) of D-digit radix=R=2J+1 nonnegative integers, with all digits $x_0$, $x_1$,..., $x_{D-1}$, lying in {0,1,2,...,J}, such that $\sum_{0\le k<D}F_J(x_k)=V$.

Claim: if a<b<c all are members of Tbh(D,R,V), then it is impossible for a+c=2b. (Proof: "carries" impossible; strictly-convex surface.)

Consequence: If X equals 1 plus the difference between the maximum and minimum elements of Tbh(D,R,V), then SalemSpencer(X)≥#Tbh(D,R,V).

Note the max and min determining X both are easy to find using "rightward-traveling greed." And we can print out the whole set #Tbh(D,R,V) for the best (cardinality-maximizing) value of V (and find that V) algorithmically in near-linear time, namely $O([J+1]^D)$ operations, because: There are only $F_J(J)D+1$ possible values of V and all the Tbh(D,R,V), for all possible V simultaneously, can be enumerated in $O([J+1]^D)$ operations. When J is odd the expected value of V is (J+2)JD/6; when J is even (J+2)(J+1)JD/(6J+6). Furthermore, if you do not want to *generate* Tbh(D,R,V) – you merely want to *estimate its cardinality* (for every V yielding high cardinality simultaneously) – with relative error≤ε (with correctness probability≥1-δ) then you can do that in $O[J^2D^{1/2}|\log\delta|\varepsilon^{-2}]$ steps by Monte Carlo. So I certainly do not agree with claims (e.g. Moser 1953) that Behrend's method is "nonconstructive." Empirically, triangular Behrend always seems either to outperform or tie original Behrend, if so completely obsoleting it.

Szekeres, Ruzsa, and my base 3, 5, and 7 methods all arise as special cases of my Tbh construction with R=3, 5, and 7. Empirically, the Tbh construction works better when R mod 4=3 (odd J) than for R mod 4=1 (even J).

**New "multiple sphere" improvements of the Beh and Tbh constructions:** Here are the cases with **2 spheres**:

- $\mathrm{Beh}_{\neq 0}$(D,R,V-1)∪Beh(D,R,V), where the "≠0" subscript means all the digits must be nonzero has (with optimal choice of V, in limits where D,R,V all go to ∞) approximately twice the cardinality of Behrend's original bound and still works, i.e. contains no 3-term arithmetic progressions.
- If J is odd then consider the set $\mathrm{Tbh}_{\neq 0,|\mathrm{distinct}|}$(D,R,V-1)∪Tbh(D,R,V), where the "|distinct|" subscript means that no two digits a,b are permitted to have |2a-J|=|2b-J| while the "≠0" subscript means that no digit a obeys $F_J(a)=0$. This set has (with optimal choice of V, in limits where D,R,V all go to ∞) approximately twice the cardinality of my Tbh bound (hence about 4 times Behrend's original bound) and still works.

It is possible to adjoin more spheres while imposing more conditions on the digits; but the conditions become more and more onerous the more spheres we adjoin, causing the resulting bounds to be less and less "constructive." Here are the cases with **3 spheres**:

- $\mathrm{Beh}_{\neq 0,|\mathrm{distinct}|}$(D,R,V-2)∪$\mathrm{Beh}_{\neq 0}$(D,R,V-1)∪Beh(D,R,V), where V≥1, and "|distinct|" here means any two digits a and b must have |a|≠|b|.
- (For J odd) $\mathrm{Tbh}_{\neq C3,|\mathrm{distinct}|,\neq 0}$(D,R,V-2)∪$\mathrm{Tbh}_{|\mathrm{distinct}|,\neq 0}$(D,R,V-1)∪Tbh(D,R,V), where V≥2, "|distinct|" here means any two digits a and b must have |2a-J| ≠|2b-J| where 2J+1=R, and "C3" means that any three digits a,b,c must have $F_J(a)+F_J(b)=F_J(c)$.

For the general case with L concentric spheres from $\sum F_J(x_k)\in\{V,V-1,...,V+1-L\}$, you get asymptotically L times greater lower bounds than plain (L=1) Beh or Tbh would have provided. [Call those sets "$\mathrm{Tbh}_L$."] This is valid for any fixed L, as well as if we permit L to grow like a sufficiently small positive constant times $(\lg X)^{1/2}/\lg\lg X$. However, the computational effort required to verify that a given D-digit number obeys the conditions, grows with L, perhaps something like $D^{\lceil L/2 \rceil}$.

Furthermore, the L-sphere Tbh idea only can significantly improve SalemSpencer lower bounds if $D \gtrapprox R^L$. That does not seem to happen, not even for L=2, within the domain covered by my table.

**Elkin**'s improvement is: instead of making Behrend's D digits, regarded as an integer D-vector, be the lattice points on the *sphere* [this sphere has radius=$V^{1/2}$ and center (0,0,...,0) in Behrend's original construction as I described it] and within the axis-oriented hypercube of sidelength=2J+1; make this set of vectors be the *extreme points of the convex hull* of the lattice points *within* both that sphere and hypercube. That (he showed) with optimal choice of V improves Behrend's bound-formula by a further factor of order$\geq(\lg X)^{1/2}$. For odd J, Elkin's idea also works for my Tbh variant; the surfaces still are spheres, but the radius now is $[2V+¼]^{1/2}$ and the center is (J/2, J/2, ..., J/2), which now is not a integer-lattice point at all. (For even J the Tbh surfaces are non-spherical, but still strictly convex.) Unfortunately Elkin's sets are algorithmically much harder to compute than the Beh and Tbh sets and their few-sphere versions. So the epithet "nonconstructive" is much more appropriate for Elkin.

Nevertheless, we now – for the first time – present an **algorithm** to produce non-averaging sets which actually can be larger than, and always are at least as large as, Elkin's extreme-point sets, and that can be implemented to run in time and memory both bounded by $|S|^{1+o(1)}$ where |S| is the cardinality of the output set. (It also permits J to be even, in which case it outputs new kinds of sets):

1. Let S(v) denote the set of D-digit radix-R numbers with all digits $x_k$ in [0,J] where R=2J+1 and obeying $\sum_{0\leq k<D} F_J(x_k)=v$.
2. Let S=S(V).
3. For k=1,2,...,$V^{o(1)}$ {

    S←S∪S(V-k). Use the prior fast algorithm to determine all b∈S that are midpoints of 2 other set elements. Then remove all such midpoints.

   }

Elkin pointed out to me that his SODA 2010 paper contained a fast algorithm for constructing his sets omitted in his 2011 journal paper. Elkin's algorithm is different from mine but perhaps comparably fast. Elkin's sets were extreme points of a convex body in D-dimensional space and hence contained, not merely "no 3-term arithmetic progressions" but actually "no $\vec{a},\vec{b},\vec{c}$ with $\vec{b}$ being *any* convex rational-linear combination of $\vec{a}$ & $\vec{c}$." Since the algorithm above only asks for the former, weaker, demand that $\vec{a}+\vec{c}\neq 2\vec{b}$, it ought to produce larger sets than Elkin, perhaps even larger by a factor which grows unboundedly with X.

**Future work.** By more prolific use of product constructions, mixed-radix generalizations of Tbh, and also considering constructions something like Elkin's and/or adjoining more set-members to Tbh sets, one could produce a larger, finer-grained, and more-precise table of lower bounds. It also would be feasible to extend it up to, say, $X=10^{999}$. Nobody seems to have decent explicit upper bounds, so somebody should try to produce them.

### "Baby Coppersmith-Winograd": E≤ln(4000/27)/ln(8)≈2.403632260832873

As an application of the Laser method, plus new ideas, Coppersmith & Winograd 1990 in their §6 devised a matrix multiplication method with exponent E≤ln(4000/27)/ln(8)≈2.40363. I'll now try to summarize the key parts of their derivation.

They begin (in their EQ5) by describing a certain task T (which is already 3-way symmetric, unlike Strassen's T we discussed previously) that they can approximately solve via a degree-3 APA ($\mathrm{APA}_3$) formula (which they state) employing q+2 multiplications. Then they take the $3P^{th}$ tensor power $T^{\otimes 3P}$ of T, which therefore is an $\mathrm{APA}_{9P}$ formula. They plan to consider large P. Let Y=2X+1 where $X=(2P)!P!^{-2}$. They will use a Z=SalemSpencer(X) set in their construction, i.e. a Z-element subset of {1,2,3,...,X} without any 3-element arithmetic progressions. They also will use Behrend's theorem that $Z\geq X^{1-o(1)}$ when X→∞.

They then create 3P+1 random quantities, each an integer in the interval [0,Y), that they call "$w_j$." They then argue that their tensor product after a certain processing based on the $w_j$'s (which does not include any bilinear muls) is applied to it, contains at least H different encrypted square-matrix products (with disjoint variables) of form (N,N,N), where $N=q^P$, where Expectation(H) ≥ $(Z/4)\,(3P)!P!^{-3}\,Y^{-2}$. (The expectation is over the randomness in the $w_j$'s; this is their EQ8.) They then argue that some suitable particular values for all their $w_j$'s *must exist* that cause H≥Expectation(H). They select and use those particular suitable values.

Therefore: if $N=q^P$ then

$$\mathrm{Rk}_{9P}[\text{ H independent copies of (N,N,N) matmul task }] \leq (q+2)^{3P}$$

for some H obeying

$$H \geq 4^{-1}\,(3P)!P!^{-3}\,\mathrm{SalemSpencer}(\,(2P)!P!^{-2}\,)\,[2(2P)!P!^{-2}+1]^{-2}.$$

Consequently,

$$\mathrm{Rk}[\text{ H independent copies of (N,N,N) matmul task }] \leq (9P+1)\,(q+2)^{3P}.$$

They argue that in the limit of large P, where the SalemSpencer "o(1)" goes to 0, this yields matrix multiplication exponent E ≤ log_q( 4 $(q+2)^3$ / 27 ) which if q=8 (the E-minimizing choice) shows E ≤ log_8(4000/27) < 2.40364.

A problem with understanding Coppersmith-Winograd is that the SalemSpencer(X) function is *unknown*. That makes it almost impossible to determine the breakeven $N_{obv}$ and $N_{str}$. All we can do is determine **upper bounds** on them by using known lower bounds on SalemSpencer(X) for appropriate X. If those lower bounds are assumed/hoped to be nearly tight, then our estimates of $N_{obv}$ and $N_{str}$ will be fairly accurate.

**Numerical Example: q=8, P=11:** Then $N=8^{11}=2^{33}=8589934592$. The combined multiplication count to multiply all H pairs of N×N matrices is

$$\#muls \le (9P+1)\,(q+2)^{3P} = 10^{35}.$$

Here the number of copies H obeys

$$H \ge 4^{-1}\ 33!\ 11!^{-3}\ \mathrm{SalemSpencer}(\ 22!\ 11!^{-2}\ )\ [22!11!^{-2}2+1]^{-2} = 34131748865760\ \mathrm{SalemSpencer}(705432)\ 1410865^{-2} \ge 34131748865760 \cdot 8320 \cdot 1410865^{-2} > 142662.867.$$

Hence

$$\#muls\ \textit{per copy} \le 10^{35} / 142662.867 \le 7.01\times 10^{29}$$

which has failed to be cheaper than the Obvious method's mul-count $N^3=2^{99}\approx 6.34\times 10^{29}$. However it would be cheaper *if* somebody could improve the lower bound on SalemSpencer(705432) from 8320 to 9202, which might be possible.

**Numerical Example: q=8, P=12:** Now $N=8^{12}=2^{36}=68719476736$. The combined multiplication count to multiply all H pairs of N×N matrices is

$$\#muls \le (9P+1)\,(q+2)^{3P} = 109\times 10^{36}.$$

Here the number of copies H obeys

$$H \ge 4^{-1}\ 36!\ 12!^{-3}\ \mathrm{SalemSpencer}(\ 24!\ 12!^{-2}\ )\ [24!12!^{-2}2+1]^{-2} = 846182940630300\ \mathrm{SalemSpencer}(2704156)\ 5408313^{-2} \ge 846182940630300 \cdot 20800 \cdot 5408313^{-2} \ge 601733.187.$$

Hence

$$\#muls\ \textit{per copy} \le 109\times 10^{35} / 601733.187 \le 1.82\times 10^{32}$$

which is cheaper than the Obvious method's mul-count $N^3=2^{108}\approx 3.25\times 10^{32}$. Hence $N_{obv}=2^{36}=68719476736\approx 6.87\times 10^{10}$ *if* our table of lower bounds were all that were known about the SalemSpencer function. But with future improvements it is possible this might drop to $N_{obv}=2^{33}=8589934592\approx 8.59\times 10^{9}$.

**Numerical Example: q=8, P=24:** Coppersmith-Winograd with $N=8^{24}=2^{72}$ uses about 37% more muls than Strassen's $7^{72}$ if the lower bound $\mathrm{SalemSpencer}(32247603683100)\ge 1.466\times 10^{9}$ is all we know.

**Numerical Example: q=8, P=25:** Coppersmith-Winograd with $N=8^{25}=2^{75}$ beats Strassen's mul-count $7^{75}$ if the lower bound $\mathrm{SalemSpencer}(126410606437752)\ge 3.69\times 10^{9}$ is used. This lower bound cannot be read directly from our bounds table but is deducible from $\mathrm{SalemSpencer}(143049436031253)\ge 4.830\times 10^{9}$ if you believe that the rightmost 17261312845878 among the 143049436031253 contain at most $1.140\times 10^{9}$ elements of the nonaveraging set. Therefore $N_{str}\le 2^{75}\approx 3.7779\times 10^{22}$.

### "Toddler" Coppersmith-Winograd: E≤2.3871900

Toddler, from §7 of Coppersmith & Winograd 1990, is conceptually similar to Baby.

In their EQ10 they invent a certain new task T (again already 3-way symmetric) that they can approximately solve via a stated $APA_3$ formula again employing q+2 multiplications. They take the $3P^{th}$ tensor power $T^{\otimes 3P}$ of T, therefore an $APA_{9P}$ formula. Let $L=\lfloor \beta P \rfloor$ where β is a constant with 0<β<1 whose optimal value they will determine later. Let Y=2X+1 where $X=(P+L)!(2P-2L)!L!^{-2}(P-L)!^{-3}$. They again use a Z=SalemSpencer(X) set in their construction, i.e. a Z-element subset of {1,2,3,...,X} without any 3-element arithmetic progressions and again use Behrend's theorem that $Z\ge X^{1-o(1)}$ when X→∞. Then again using a "randomized encryption" scheme they argue that their tensor product after a certain "decryption" contains at least H different encrypted square-matrix products (with disjoint variables) of form (N,N,N), where $N=q^{P-L}$, where

$$\mathrm{Expectation}(H) \ge (Z/4)\ (3P)!(P-L)!^{-3}L!^{-3}\ Y^{-2}.$$

They then again argue that some suitable particular values for all their randoms *must exist* that cause H≥Expectation(H), and select and use those particular suitable values. Therefore: if $N=q^{P-L}$ then

$$\mathrm{Rk}_{9P}[\text{ H independent copies of (N,N,N) matmul task }] \le (q+2)^{3P}$$

for some H obeying

$$H \ge 4^{-1}\ (3P)!(P-L)!^{-3}L!^{-3}\quad \mathrm{SalemSpencer}(\ (P+L)!(2P-2L)!L!^{-2}(P-L)!^{-3}\ )\quad [2(P+L)!(2P-2L)!L!^{-2}(P-L)!^{-3}+1]^{-2}.$$

Consequently,

$$\mathrm{Rk}[\text{ H independent copies of (N,N,N) matmul task }] \le (9P+1)\,(q+2)^{3P}$$

which is $\le(9P+1)(q+2)^{3P}/H$ muls *per copy*. They argue that in the limit of large P, where the SalemSpencer "o(1)" goes to 0, if they take q=6 and β=6/125=0.048, this yields matrix multiplication exponent E≤2.38719. Notice that with this choice of β, we already need P≥21 just to allow L>0 so that this method can operate nontrivially at all.

**Numerical Example: q=6, P=21, L=1:** This P is the minimum allowed causing Toddler to be a nontrivial algorithm, i.e. to have L>0. Then $N=6^{20}$=3656158440062976. The combined multiplication count to multiply all H pairs of N×N matrices is

$$\#\text{muls} \le (9P+1)\,(q+2)^{3P} \approx 190\times 8^{63} \approx 1.491\times 10^{59}.$$

Here the number of copies H obeys

$$H \ge 4^{-1}\, 63!\, 20!^{-3}\, 1!^{-3}\, \text{SalemSpencer}(\, 40!\, 22!\, 1!^{-2}\, 20!^{-3}\,)\, (40!22!1!^{-2}20!^{-3}2+1)^{-2} = 3441938303723213016028084013235\text{0 SalemSpencer}(63685096314840)\, 127370192629681^{-2}$$

Using SalemSpencer(63685096314840)>$3.273\times 10^{9}$ we find H ≥ $6.944\times 10^{12}$. Hence

$$\#\text{muls } \textit{per copy} \le 1.491\times 10^{59} / (6.944\times 10^{12}) \le 2.147\times 10^{46}$$

which is cheaper, by more than a factor of 2, than the Obvious method's mul-count $N^3=6^{60}\approx 4.8874\times 10^{46}$. Hence $N_{obv}=6^{20}$=3656158440062976 with $\log_2(N)\approx 51.69925$.

**Numerical Example: q=6, P=32, L=1:** Then $N=6^{31}$=1326443518324400147398656. The combined multiplication count to multiply all H pairs of N×N matrices is

$$\#\text{muls} \le (9P+1)\,(q+2)^{3P} \approx 289\times 8^{96} \approx 1.437\times 10^{89}.$$

Here the number of copies H obeys

$$H \ge 4^{-1}\, 96!\, 31!^{-3}\, 1!^{-3}\, \text{SalemSpencer}(\, 62!\, 33!\, 1!^{-2}\, 31!^{-3}\,)\, (62!33!1!^{-2}31!^{-3}2+1)^{-2} = 445908090855572846831392535773770961755140997120\, \text{SalemSpencer}(491492341037555708928)\, 982984682075111417857^{-2}$$

Using SalemSpencer(491492341037555708928)>$7.333\times 10^{14}$ we find H ≥ $3.384\times 10^{20}$. Hence

$$\#\text{muls } \textit{per copy} \le 1.437\times 10^{89} / (3.384\times 10^{20}) \le 4.247\times 10^{68}.$$

This fails to be as cheap as Strassen's mul-count $M_{str1}(N)$=647905355439564758734710630362670339005681570353239722895222375084 92≈$6.479\times 10^{67}$. Hence $N_{str}>6^{31}$ if our lower bound on SalemSpencer(491492341037555708928) is correct to within a factor of 6.

Similarly we find **P=37** also is not enough to make Toddler beat $M_{str1}(N)$, at least if the conjectural bound SalemSpencer(62217263162923251156082 4)>$1.6\times 10^{17}$ is not too weak. However, **P=38** suffices to make Toddler beat $M_{str1}(N)$, indeed by over 20%, using SalemSpencer(25877654963453980429710 24)≥$4.706\times 10^{17}$. Hence $N_{str}\le 6^{37}\approx 6.189\times 10^{28}$ with $\log_2(N)\approx 95.6436$.

## "Monster" Coppersmith-Winograd: E≤2.3754770

Monster, from §8 of Coppersmith & Winograd 1990, is conceptually similar to Toddler. However, they now use intentionally-*coupled* (rather than statistically independent) uniform random variables in their decryption scheme; they begin by defining T now to be the *tensor square* of (their EQ10) Toddler task; and they do not merely find a lot of independent copies of square-matrix multiplications inside the high tensor powers of T they then consider, but rather a variety of different sizes and shapes of rectangular matrix multiplication tasks – then they use Schönhage's ASI on those. They also need to optimize three real parameters simultaneously to tune their construction (whereas for Toddler there was only one real parameter "β" to optimize).

I'm not going to try to work out the breakeven N for Monster because I do not understand it well enough. I think, though, that they considerably exceed the corresponding breakeven N's for Toddler. It took everybody 20 years to recover from Coppersmith-Winograd 1990, but then during 2010-2023, other workers

> **Authors of papers on "post-Coppersmith-Winograd" fast matrix multiplication:**
> A.M.Davie, Andrew J. Stothers, Virginia Vassilevska-Williams, Francois Le Gall, Josh Alman, Ran Duan, Hongxun Wu, Renfei Zhou

considered higher tensor powers (up to power 32) in CW's E=2.375477 construction, improved the randomized encryption/decryption schemes, and/or added other tricks, while solving optimization problems (or trying to) sometimes with over 300 real parameters(!) to find out how to optimally tune their algorithms... which allegedly improved the exponent bound even further to E≤2.3736898 and E≤2.3729269 and E≤2.3728639 and E≤2.3728596 and E≤2.371866. These later schemes make Monster look simple by comparison, and I believe their breakeven N's exceed Monster's. I shall not examine any of them either. However, I think that at least some of the authors who devised those improvements must have written special purpose software to help them; and that software hopefully could be repurposed comparatively easily, to estimate their breakeven N's. Unfortunately when I tried enquiring about that by emailing those authors, I received zero responses.

Coppersmith-Winograd (baby, toddler, and monster) all can be described (or criticized) as seeming closer to a "**nonconstructive** proof that an algorithm exists" than "an algorithm" for matrix multiplication.

## The rate of decrease of exponents (for true op-counts that are *not* power-laws)

For the simplest recursive matrix-multiplication schemes, e.g. based on Strassen's 7-mul (2,2,2) formula and Smirnov's 40-mul (3,3,6) formula, the mul-count of the algorithm applied to N×N matrices is bounded between two positive constants times $N^E$ (for whatever E is appropriate for that algorithm) for all N≥1. However, for plain-APA schemes, e.g. based on Smirnov's 20-mul $APA_6$ formula for (3,3,3), that is not true! Now the mul-count is bounded between two positive constants times $N^E\log(N)$. Equivalently we can regard this as $N^{E(N)}$ with *non-constant* exponent $E(N)=(c+\ln\ln N)/\ln N+E(\infty)$ for some c bounded between two constants. In other words the exponent E(N) now is **decreasing** toward its limit value – and rather slowly!

The mul-count of the Schönhage ASI scheme (with E≈2.547993) we analysed is bounded between two positive constants times $N^E \log(N)^{3/2}$, which we may equivalently regard as $N^{E(N)}$ where $E(N)=(3/2)(c+\ln\ln N)/\ln N+E(\infty)$ for some c bounded between two constants.

With the particular instantiation of Strassen's laser method we examined (q=8, E≈2.478495), the mul-count is bounded between two positive constants times $N^E \log(N)^3$. We may equivalently regard this as $N^{E(N)}$ where $E(N)=3(c+\ln\ln N)/\ln N+E(\infty)$ for some c bounded between two constants.

With Coppersmith-Winograd, an additional ingredient enters the pot: the Behrend/Elkin lower bounds on the SalemSpencer(X) function. These (if presumed nearly tight) cause E(N) for "Baby" to behave like $E(N)=4(3\lg N)^{-1/2}+E(\infty)$, and for "Toddler" like $E(N)=(8.00689/\lg N)^{1/2}+E(\infty)$. ?? ignoring lower order terms. These are much slower rates of decrease for E(N). E.g, back in the fixed-E point of view Baby's mul-count is $N^E \exp(4[\ln(2^{1/3})\ln(N)]^{1/2}[1\pm o(1)])$.

To give you an idea of just how slow these rates of approach to the limit E(∞) are, we tabulate crude estimates of the least N needed to cause E(N) to approximate E(∞) accurate to 1, 2, and 3 decimal places:

| Quantity | N causing Quantity=0.1 | N causing Quantity=0.01 | N causing Quantity=0.001 | Comment |
|---|---|---|---|---|
| lnlnN/lnN | $3.43\times10^{15}$ | $1.29\times10^{281}$ | $7.94\times10^{3959}$ | plain APA schemes |
| (3/2)lnlnN/lnN | $7.46\times10^{26}$ | $5.07\times10^{452}$ | $4.42\times10^{6235}$ | Certain Schönhage ASI schemes including the one with E≈2.548 we examined |
| 3lnlnN/lnN | $2.08\times10^{65}$ | $9.02\times10^{1009}$ | $1.81\times10^{13475}$ | Certain Strassen Laser schemes including the one with E≤2.479 we examined |
| $4(3\lg N)^{-1/2}$ | $3.54\times10^{160}$ | $8.57\times10^{16054}$ | $2.04\times10^{1605493}$ | Coppersmith-Winograd "Baby" (Behrend) |
| $(8.00689/\lg N)^{1/2}$ | $1.07\times10^{241}$ | $1.38\times10^{24103}$ | $1.15\times10^{2410314}$ | Coppersmith-Winograd "Toddler" (Behrend) |

Evidently, when authors of theoretical papers helpfully state exponents E accurate to 10 decimal places... that is not always as helpful as they think. The final line of the table also suggests that in order for Coppersmith-Winograd Toddler to beat Schönhage's ASI scheme with E≈2.548, we need N at least $10^{90}$.

## The limitations of asymptotics in theoretical computer science

Once an algorithm's breakeven N's correspond to input sizes exceeding the number of bits of entropy storable in the **observable universe**, which should be below $10^{124}$ (from Bekenstein-Hawking entropy of black hole with mass=$10^{53}$kg=mass in observ.universe) then we can agree it is no longer going to be an algorithm of practical interest. (Actually, black holes seem to be "write only memories," posing a slight problem. If we agree to abstain from using black holes then the entropy potentially storable using the particles present in the observable universe today – mainly photons, neutrinos, and gravitons left over from the big bang – should be below $10^{91}$ bits.) That is: the number $3N^2$ of matrix entries in the 3 matrices exceeds $10^{124}$ when N exceeds $6\times10^{61}$, or equivalently $\log_2(N)>205.2$.

This threshold definitely is exceeded for the N causing breakeven between Bini et al's (2,2,3) APA based scheme versus Strassen's 2×2-based scheme. It also seems exceeded for the N causing breakeven between Coppersmith-Winograd "Toddler" versus the Schönhage ASI scheme with E≈2.548.

Those of us who lack confidence in our ability to command all information potentially storable in the entire observable universe, might regard it as beyond reach if a problem's input size, in bits, exceeds merely the number $2\times10^{50}$ of atoms in **planet Earth**. This lesser threshold seems safely exceeded by the $N_{str}$ causing breakeven between Coppersmith-Winograd Toddler, versus Strassen's 2×2 method.

To place the Universe and Earth thresholds in perspective, let us mention a few other problems. John Tromp upper bounded the number of legal positions (including en passant and castling statuses and side to move) in **chess** by $8.727\times10^{45}$ (and $2.892\times10^{39}$ for positions with no promoted pawns). He estimated the true number to be $4.82\times10^{44}$ to within ±10%. That would pose no obstacle to a demigod who could control the isotopic type of each atom in the dwarf planet Ceres ($10^{46}$ atoms) – such a demigod could solve chess and store the win/loss/draw value of every legal posiiton.

Now instead consider the oriental game of **go** which is played on an H×W grid (traditionally H=W=19) with each grid point either occupied by a black stone, a white stone, or empty. The number of go position diagrams therefore is $3^{361}$. However, only a subset of these can actually arise during a legal game of go. The exact cardinality T of that subset was computed by Tromp & Gunnar Farnebäck in 2016: $T\approx2.081681994\times10^{170}\approx3^{356.97}$. Unfortunately, a hypothetical database giving the perfect-play win/loss value (or final score value; you win if your final score, minus your opponent's, exceeds the pre-agreed "komi" constant value) of every such position would *not* necessarily solve go. That's because of the "superko rule" that moves that repeat a go-diagram that previously occurred in the game are illegal. This rule causes the true game-state *not* to merely be described by the board diagram plus "who is to move" – also a possibly-huge amount of game prior-history information could be required! Although the superko rule prevents go games from lasting for an infinite number of moves, it does not stop them from lasting for at least $10^{100}$. This problem could be greatly reduced if we added to the Tromp-Taylor ruleset for go, this additional rule:

> If any string of 100 consecutive moves occurs whose net effect is *not* to decrease the number of empty grid points, then the game instantly ends. (I do not know whether "100" is the best value to use.)

This would ensure that go games last <36100 moves. Tromp suspects that "multiple ko" situations do not occur in perfect play – although triple-to-quintuple kos *have* occurred in professional go games at a historical rate of about once per 8000 games – in which case a database size below 361T presumably indeed would solve go.

Anyhow, for us the important thing is that such a T-bit go database would be too big to be storable in the observable universe.

As our final example, consider the Harvey & van der Hoeven 2021 algorithm for multiplying N-bit integers in time O(NlogN) on a multitape Turing machine (which time-bound conjecturally is optimal). The crucial step in this algorithm which distinguishes it from previous slower ones can only happen when $N\geq2^{1729}$.

When I was taught computer science as a student in 1980s my teachers contended that only asymptotic bounds matter; so algorithms with better asymptotics are better – for all but a tiny number of atypically-naughty algorithm examples. Unfortunately it looks now that the reality is more the opposite. The term "galactic algorithm" was introduced by K.Regan & R.J.Lipton to denote the "rare" naughty examples, which in fact appear to be the majority of algorithms at CS theory conferences today.

### Conclusions

Strassen-like exact formulas and perhaps some instantiations of Schönhage's ASI are the only two fast matrix multiplication ideas yet proposed with breakeven N small enough to possibly have practical interest. These are the rows color-coded white and yellow in our main table.

### References


Felix A. Behrend: On sets of integers which contain no three terms in an arithmetic progression, Proc. Nat. Acad. Sci. USA 32,12 (Dec.1946) 331-332.

Dario A. Bini, Milvio Capovani, Francesco Romani, Grazia Lotti: $O(n^{2.7799})$ Complexity for n×n Approximate Matrix Multiplication, Information Processing Letters 8,5 (June 1979) 234-235.

Thomas F. Bloom & Olof Sisask: The Kelley-Meka bounds for sets free of three-term arithmetic progressions, Essential Number Theory 2,1 (2023) 15-44. and An improvement to the Kelley-Meka bounds on three-term arithmetic progressions (both 2023).

B.Boyer, J.-G.Dumas, C.Pernet, W.Zhou: Memory efficient scheduling of Strassen-Winograd's matrix multiplication algorithm, Proc. 2009 International Symposium on Symbolic & Algebraic Computation (ISSAC 2009) 55-62. Their table 10 summarizes their algorithms; they implemented and tested several on an Intel "3GHz core2 duo" finding they all outperformed Obvious when N=4096.

Robert Breusch: Zur Verallgemeinerung des Bertrandschen Postulates, dass zwischen x und 2x stets Primzahlen liegen, Mathematische Zeitschrift 34 (1932) 505-526.

Kevin S. Brown: No Four Squares In Arithmetic Progression web page. The most readable proof I found was a blatantly plagiarized, then streamlined and republished, version of Brown's proof by Alfred van der Poorten (2007) at http://arxiv.org/pdf/0712.3850.pdf.

Keith Conrad: Arithmetic Progressions of Four Squares, `http://kconrad.math.uconn.edu/blurbs/ugradnumthy/4squarearithprog.pdf`.

Don Coppersmith: Rectangular matrix multiplication revisited, J. of Complexity 13,1 (Mar.1997) 42-49.

Don Coppersmith & Shmuel Winograd: On the asymptotic complexity of matrix multiplication, SIAM J. Comput. 11,3 (1982) 472-492.

Don Coppersmith & Shmuel Winograd: Matrix multiplication via arithmetic progressions, Proceedings 19th annual ACM conference on Theory of computing STOC (1987) 1-6.

Don Coppersmith & Shmuel Winograd: Matrix multiplication via arithmetic progressions, J.Symbolic Computation 9,3 (Mar.1990) 251-280.

Henri Darmon & Loic Merel: Winding quotients and some variants of Fermat's Last Theorem, J. Reine Angew. Math. 490 (1997) 81-100.

J.Demmel, I.Dumitriu, O.Holtz: Fast Linear Algebra Is Stable, Numerische Mathematik 108,1 (2007) 59-91.

J.Demmel, I.Dumitriu, O.Holtz, R.Kleinberg: Fast Linear Algebra Is Stable, Numerische Mathematik 106,2 (2007) 199-224.

Peter Denes: Über die Diophantische Gleichung $x^\ell+y^\ell=cz^\ell$, Acta Math. 88 (1952) 241-251.

Leonard E. Dickson: History of the theory of numbers, 3 vols, Chelsea reprint 1966. QA241.D5.

C.C.Douglas, M.Heroux, G.Slishman, R.M.Smith: GEMMW: A portable level 3 BLAS Winograd variant of Strassen's matrix-matrix multiply algorithm, J.Computational Physics 110 (1994) 1-10.

Ch.-E.Drevet, Md.Nazrul Islam, E.Schost: Optimization techniques for small matrix multiplication, Theoretical Computer Science 412,22 (May 2011) 2219-2236.

Michael Elkin: An improved construction of progression-free sets, Israel J. Math. 184 (2011) 93-128. A considerably shorter, but unfortunately even "less constructive," proof of (essentially) Elkin's result was found by B.Green & J.Wolf in 2008. Elkin also pointed out to me that he'd also written a conference version: Proc. Annual ACM-SIAM Symposium on Discrete Algorithms SODA 21 (2010) 886-905 which contains some material omitted in the journal version.

Pal Erdös: Über die Primzahlen gewisser arithmetischer Reihen [On the prime numbers of certain arithmetic series], Mathematische Zeitschrift 39 (1935) 473-491. Similar results were proven by similar methods by Giovanni Ricci in Bolletino della Unione Matematica Italiana 1933-1934.

William Gasarch, James Glenn, Clyde P. Kruskal: Finding large 3-free sets I: The small n case, J. of Computer and System Sciences 74,4 (June 2008) 628-655.

William Gasarch, James Glenn, Clyde P. Kruskal: Finding Large Sets Without Arithmetic Progressions of Length Three: An Empirical View and Survey II (2010 preprint, never published, except on internet: `http://www.cs.umd.edu/~gasarch/BLOGPAPERS/3apsurvey.pdf`. Appendix VII contains errors, for example in the line beginning $10^{13}$ the claimed ordering of B3 and B5 differs from what their numbers say. In appendix IV, many "recommended" values in table 5 are incorrect. I sent GG&K a counterexample indicating their "SPHERE_NZ" bound and their theorem 38 (which seems the main new result claimed) are incorrect. Therefore I presume lemma 35.1, lemma 35.4, and the claim in proof of theorem 38 that "we can ignore the cases where c=1" all are incorrect.

Tor Hadas & Oded Schwartz: Towards Practical Fast Matrix Multiplication based on Trilinear Aggregation, ISSAC '23: Proceedings of the 2023 International Symposium on Symbolic & Algebraic Computation (July 2023) 289-297.

David Harvey & Joris van der Hoeven: Integer multiplication in time O(NlogN), Ann. of Math. (2), 193,2 (March 2021) 563-617.

John E. Hopcroft & Leslie R. Kerr: On minimizing the number of multiplications necessary for matrix multiplication, SIAM J. Appl. Math. 20 (1971) 30-36.

Jean Itard: Arithmetique et Theorie des Nombres, Presses Universitaires de Paris, Paris 1963 (also was a 3rd ed, 1973).

Igor Kaporin: A practical algorithm for faster matrix multiplication, Numerical linear algebra with applications 6,8 (1999) 687-700.

Igor Kaporin: The aggregation and cancellation techniques as a practical tool for faster matrix multiplication, Theoretical Computer Science 315, 2-3 (2004) 469-510.

Elaye Karstadt & Oded Schwartz: Matrix multiplication, a little faster, J.of the ACM 67 (2020) 1-31.

Zander Kelley & Raghu Meka: Strong bounds for 3-progressions (2023), `http://arxiv.org/abs/2302.05537`.

Karl Molsen: Zur Verallgemeinerung des Bertrandschen Postulates, Deutsche Math. 6 (1941) 248-256.

Pieter Moree: Bertrand's Postulate for primes in arithmetic progressions, Computers mathematics applications 26,5 (1993) 35-43.

Leo Moser: On Non-Averaging Sets of Integers, Canadian J. Maths 5 (1953) 245-252.

Victor Ya. Pan: New fast algorithms for matrix operations, SIAM J. Computing 9,2 (May 1980) 321-342.

Victor Pan: New combinations of methods for the acceleration of matrix multiplications, Computers & Mathematics with Applications 7,1 (1981) 73-125.

Victor Pan: Trilinear aggregating with implicit canceling for a new acceleration of matrix multiplication, Computers & Mathematics with Applications 8,1 (1982) 23-34.

Victor Pan: How can we speed up matrix multiplication? SIAM Review 26,3 (July 1984) 393-415.

Victor Pan: How to multiply matrices faster, Lecture Notes in Computer Science #179, Springer, 1984.

Victor Pan: The techniques of trilinear aggregating and the recent progress in the asymptotic acceleration of matrix operations, Theoretical Computer Science 33,1 (1984) 117-138.

Leonardo Pisano ("Fibonacci"): Liber Quadratorum (1225). L.E.Sigler republished this as *The Book of Squares* ("An [Extensively] Annotated Translation into Modern English"), Academic Press Boston 1987; and Reinhard Schultz posted a redo of Fibonacci's characterization of the 3-term arithmetic progressions of squares on the internet at http://math.ucr.edu/~res/math153-2019/history07b.pdf.

Robert L. Probert: On the additive complexity of matrix multiplication, SIAM J. Comput. 5,2 (1976) 187-203.

Alexey Vladimirovich Smirnov: The bilinear complexity and practical algorithms for matrix multiplication, Computational Mathematics & Mathematical Physics, 53,12 (Dec.2013) 1781-1795, Originally published in Russian in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 53,12 (2013) 1970-1984.

Arnold Schönhage: Partial and total matrix multiplication, SIAM J. Computing 10,3 (Aug.1981) 434-455

Volker Strassen: Gaussian elimination is not optimal, Numerische Mathematik 13,4 (Aug.1969) 354-356.

Volker Strassen: Rank and optimal computation of generic tensors, Linear algebra & its applications, 52-53 (July 1983) 645-685.

Volker Strassen: Relative bilinear complexity and matrix multiplication, J. für die reine und angewandte Mathematik 375/376 (1987) 406-443.

Volker Strassen: The asymptotic spectrum of tensors, J. für die reine und angewandte Mathematik 384 (1988) 102-152.

Ondrej Sykora: A fast non-commutative algorithm for matrix multiplication, pp.504-512 in Jozef Gruska, editor, Proceedings 6th International Symposium on Mathematical Foundations of Computer Science, Springer Lecture Notes in Computer Science #53, Tatranska Lomnica, Czechoslovakia, September 1977.

Shmuel Winograd: On multiplication of 2×2 matrices, Linear algebra & its applications 4,4 (Oct.1971) 381-388.