An Algorithm for Generating Highly Composite Numbers

D. B. Siano and J. D. Siano

Oct. 7, 1994

Introduction

All positive integers can be written in only one way as a product of powers of primes:

$$n = 2^{a_2} 3^{a_3} 5^{a_5} \ldots . p^{a_p} . \qquad (1)$$

The number of distinct divisors of n is, since each power of a prime factor can include zero,

$$d(n) = (a_2 + 1)(a_3 + 1)(a_5 + 1)\ldots(a_p + 1). \qquad (2)$$

When n=12, for example, eqs. 1 and 2 yield 12 = 2^2 3^1 so d(n)= (2+1) (1+1) = 6. The divisors of 12 can be readily enumerated as {1,2,3,4,6,12}.

The divisor function d(n) has been of great interest to number theorists for a long time. It fluctuates wildly from one integer to the next, and one might think it would be quite unpredictable. However, it is actually possible to derive some simple rules about its average behavior. One result, for example, is that the average of the number of divisors of n from 1 to N is approximately ln[N].

Prime numbers, of course, have the minimum number of divisors possible: 2 (1 and the prime number itself). It is natural to examine the numbers at the other end of the range--the numbers that have the highest possible number of divisors. These numbers were first studied by S. Ramanujan and a number of interesting results were demonstrated.

Highly composite numbers

The definition of a highly composite number (integer) is that it is a number that has a larger number of divisors than any number less than itself. For example, 12 is a highly composite number, because it has 6 divisors while every number less than 12 has a smaller number of divisors: e.g. 10, 8 and 6 have 4 divisors, 9 has only 3, etc. The sequence of highly composite numbers starts out as 2, 4, 6, 12, 24, 36, 48, 60, 120, 180, 240, 360, 720, 840, 1260, etc. The 100th number published by Ramanujan is 3212537328000 which has 8192 divisors. while the average number of divisors of all the numbers up to this one is about 29.

It is easy to show that composite numbers have some interesting simple properties. For example, for n to be highly composite, none of the prime factors in eq. 1 can be omitted. Thus,

$$k = 2^4 3^6 7^3$$

cannot be highly composite (5 is omitted) because

$$k' = 2^4 3^6 5^3$$

has the same number of factors as k, by virtue of eq. 2, but is smaller than k. Also, by similar reasoning, k' cannot be highly composite either, because

$$k'' = 2^6 3^4 5^3$$

is obviously smaller than k' and has the same number of factors as k'.

Clearly, for n to be highly composite, we must have

$$a_2 \geq a_3 \geq a_5 \geq a_7 ... \geq 1.$$

Ramanujan gave a list of the first one hundred of these numbers (with one error of omission), which he found "by trial". There are several outstanding conjectures about the properties of the sequence of highly composite numbers and it would be of some interest to have a method of generating them automatically. A very simple algorithm, in the form of a sieve, is

```
n:=2;  nd:=2;
label1:  n:=n+1;
if divisors(n)≤ nd  then  goto label1;
else nd:=divisors(n), print n, goto 1;
```

will in principle print out all of the highly composite numbers. However, this is much too slow for even moderately large n and it quickly runs out of steam. Highly composite numbers are relatively rare: there are (roughly) only 10 per decade. Therefore speeding the sieve up by an order of magnitude will therefore result in only an additional 10 highly composite numbers. To calculate the thousandth highly composite number by the brute force sieve using a computer that tested one number per picosecond would still take many times (10^45 or so times!) the age of the universe.

The challenge is to discover a much faster algorithm that can yield highly composite numbers far larger than those already tabulated.

Some preliminary observations

It is instructive to examine a short section of the known highly composite numbers:

n	pwrs of primes	hc(n)/hc(n-1)
85	63221111	28/23
86	731111111	46/35
87	54221111	105/92
88	532111111	23/21
89	442111111	3/2
90	64221111	28/23
91	632111111	23/21
92	542111111	3/2
93	732111111	4/3
94	642111111	3/2
95	532211111	7/6
96	633111111	10/7
97	442211111	21/20

They are listed here in a compact way--only the powers of the successive primes are shown, e.g., for n=85, the highly composite number is

$$2^6 3^3 5^2 7^2 11^1 13^1 17^1 19^1 = 97772875200.$$

There are several things worth noting. The prime having the highest power is 2, and the power of the highest prime is unity. Ramanujan proved that this is nearly always true,with only two exceptions: 4 and 36. Unfortunately, the highest prime factor does not increase monotonically with n. The ratios of successive highly composite numbers are always rational numbers greater than one. The ratios are also easily seen to be less than or equal to two: multiplying any highly composite number by two always gives a larger number with a larger number of divisors. Thus, given a highly composite number n, we are guaranteed to always have at least one in the range between n and 2n.

The algorithm

A method for calculating successive highly composite numbers can be devised from these observations. The essential part is to get hc(n+1) from the previous one, hc(n) by multiplying it by a suitable ordered (increasing) list of rational numbers {r}, where each member of {r} is greater than one and less than or equal to two. The number of divisors of each of the products is calculated and compared to the number of divisors of hc(n); the first one with a greater number of divisors is hc(n+1).

The interesting part is to calculate a suitable list {r}. We consider first the case where the highest prime factor, p, does not change from hc(n) to hc(n+1). It is useful to factor hc(n) into two parts: a highly variable one, which we denote v, and a part that is the same for both hc(n) and hc(n+1), which is defined as

$$s = \prod_{i=1}^{m} p_i \,,$$

where m is the ordinal number of the highest prime factor. The other, more variable part, is given by

$$\mathrm{v} = \mathrm{hc(n)/s.}$$

For example for n = 85

$$s(85) = 2\ \ 3\ \ 5\ \ 7\ \ 11\ \ 13\ \ 17\ \ 19$$

and

$$\mathrm{v}(85) = 2^5 3^2 5^1 7^1$$

If a maximum largest power (of 2) is assumed to be g (say 8, for example), then it is easy to generate a list of numbers like v with the constraint that the successive powers are non-increasing:

```
do i=g to 1 step -1
do j=i to 1 step -1
do k=j to 1 step -1
do l=k to 1 step -1
v(i,j,k,l)= 2^i 3^j 5^k 7^l
end do
```

When g is not too large, nor the number of prime factors under consideration, f, (f=four in the case here) too many, the list of v's is not so large as to be unmanageable. It is not hard to show that the number of terms in v is

$$n = \frac{\prod_{i=1}^{f}(i+g)}{f!}.$$

Following the example along, when g=8 and f=4, this list has 495 terms in it. It can be narrowed to just those that are larger than v(85), and smaller than or equal to two times v(85). Multiplying each member of the list of v's by s, then ordering it gives a$_0$,list, V of numbers that are candidates for hc(n+1). In the example, this gives a list of only 11 elements.

Sometimes the largest prime factor in hc(n+1) is larger than that in hc(n). To cover this possibility a small modification of the procedure is required. The same list of v's is used, but the search is narrowed to just those that are between rp and 2 rp where rp = v(85)/pp, and pp is the next prime after p. Multiplying the list of v's by s pp will give a list Vp, which in the example given, has 7 members.

Similarly, when the largest prime factor in hc(n+1) is smaller than in hc(n), the search is narrowed to just those that a are between rm and 2 rm, where rm = v(85) pm, and pm is the prime before p. Multiplying the list of v's by s/pm will give a list Vm. For the example given it has 17 members.

Each of these three lists are then combined (Union) and has only 35 members for the example) into one list, ordered and searched for the first one having a larger number of divisors than hc(n) to give, finally hc(n+1). For the example, the correct result is found on the 13th try. Narrowing the search in this way minimizes the time and memory requirements, but is still clearly exhaustive of all of the possibilities, provided g and f are chosen to be large enough. Proper limits on these are discussed in the appendix.

The above process can then be repeated to give hc(n+2). In principle a new list of v's, with increased values for g and/or f may be assumed to get it. In practice it is better to make the list of v's large enough in the beginning to encompass the range of interest, and use the same list over and over again until done, or a larger list must be calculated. Thus f=4 and g=6 gives a list sufficiently large to calculate all of the one hundred highly composite numbers given by Ramanujan. On a PowerMac 7100/66, using the Mathematica program language, the calculation of the first 100 takes only about 10 s. The complete listing of the algorithm is given in Appendix II.

Properties of the calculated highly composite numbers

The final results of the calculation of 1000 terms in the series of highly complex numbers is shown in Fig. 1. The complete table is too long to show here, but we may note briefly that the l000th highly composite number is found to have 76 digits: it is

5073959532491205017030552999663023046456302487981340971887896279504 6826080000,

and it has 109586090557440 divisors. This can be compared to the average number of divisors up to this number, which is approximately ln[hc(1000)) = 177.

The powers of the successive primes of this number is also interesting. It can be written in a compact form:

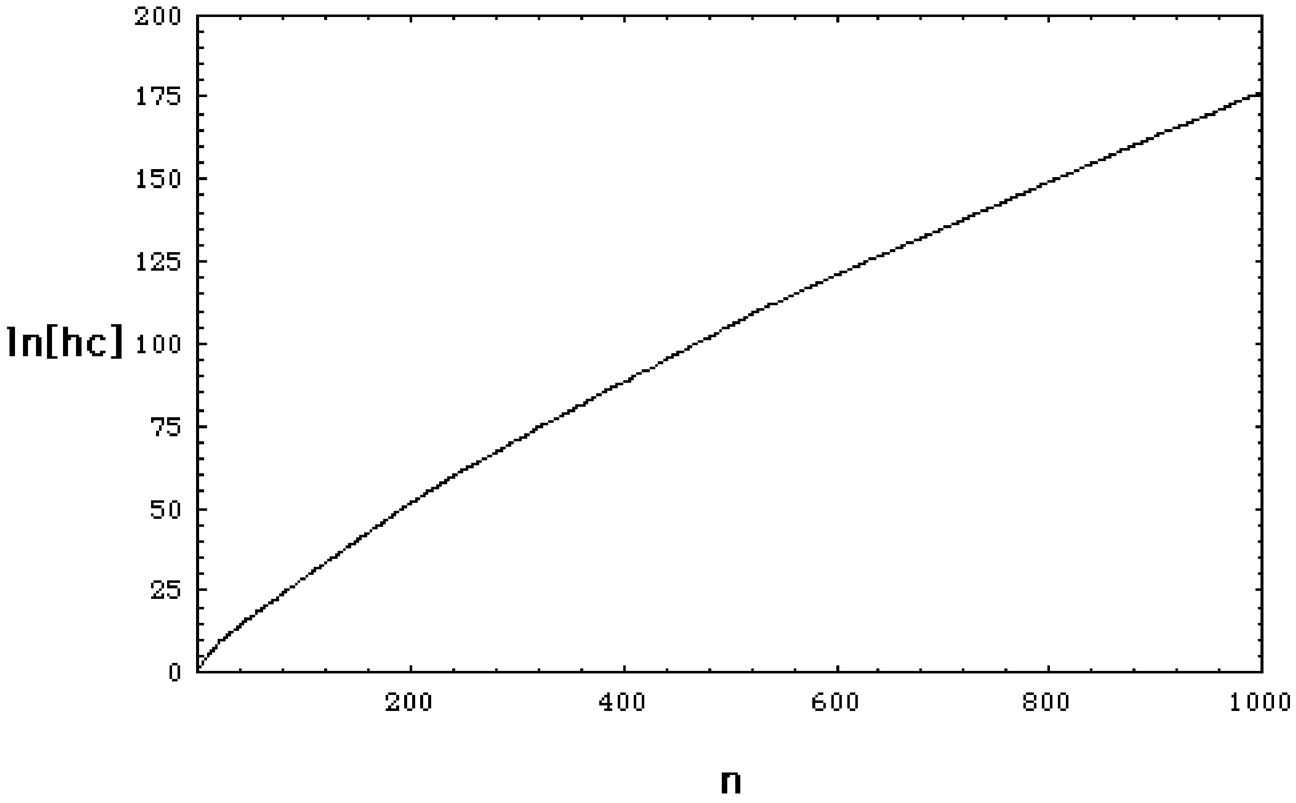
864322221111111111111111111111111111;

That is, $2^8\ 3^6\ 5^4\ 7^3$ .... 157 163.

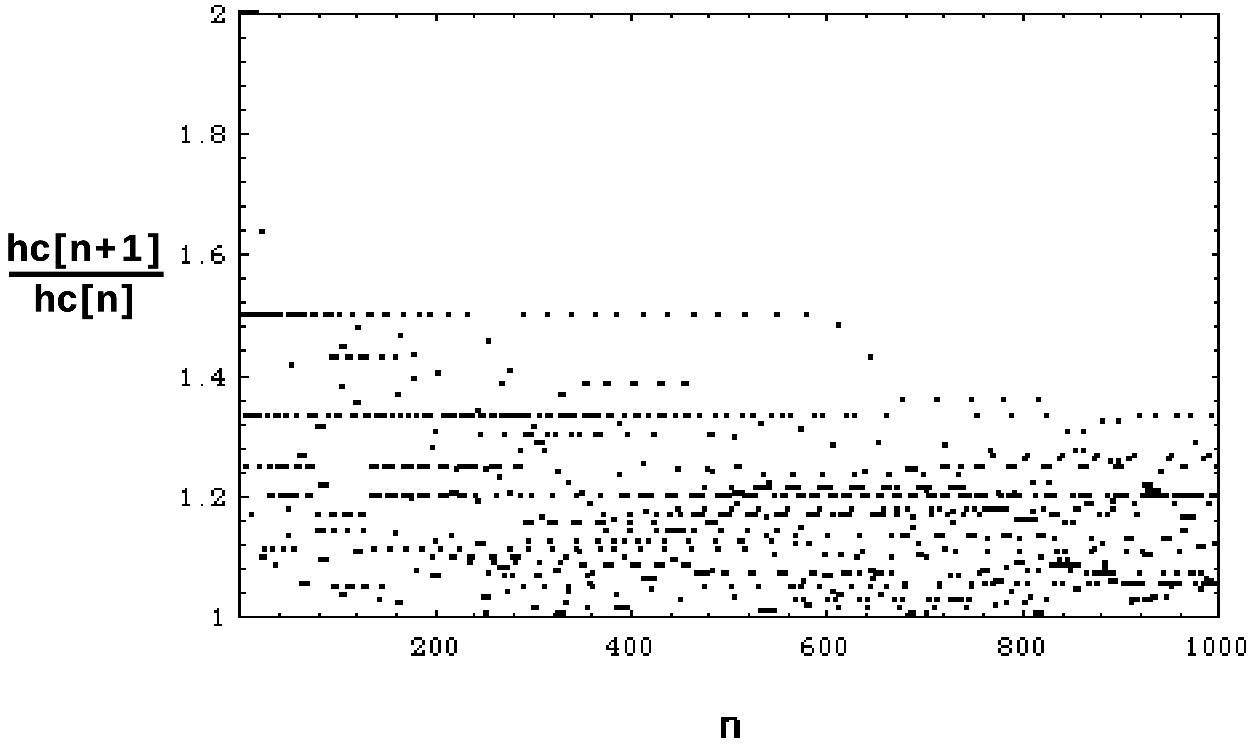
Examination of the numbers in the entire range computed shows that the first three exponents (of 2, 3 and 5) in all of the numbers larger than the 517th one are strictly decreasing. Ramanujan showed that in very large highly composite numbers, we would have $a_2 > a_3 > a_5 \ldots > a_\lambda$ (strictly decreasing exponents) when $\ln(p) > 8\lambda^2 (\ln(\lambda))^3$. For $\lambda = 5$, this would make p huge. Evidently the strictly decreasing exponents occurs far earlier than predicted by this inequality.

We can make use of this fact to decrease the size of the list of v's required for still larger numbers, and speed up the algorithm too.

The calculation of the 1000 highly composite numbers took only about 2000 seconds on the PowerMac, but took a fair amount of memory--24 M of RAM with 45 M of virtual memory. The list of v's required for the last 100 terms had over 48,000 entries.



Although the hc(n) appear on this scale to be a smooth function of n, the ratio of successive numbers, shown in Fig. 2 illustrates the variation more clearly. The ratios, of course generally decrease with increasing n, as may be expected from the decreasing slope in Fig. 1.



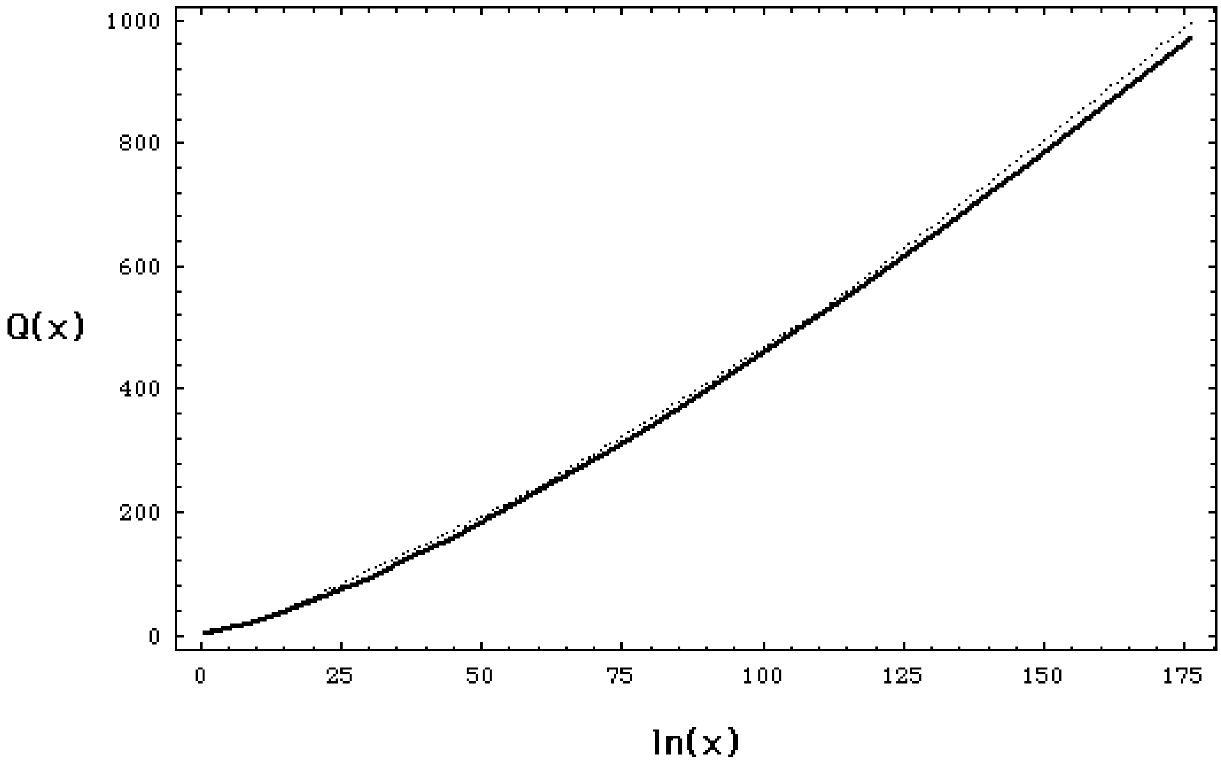
The density of the highly composite numbers is also of some interest. Here we define the number of highly composite numbers less than x to be Q[x]. For example, for z=100, Q[x]=8, and for z=1000, Q[x] = 14. Fig.1 shows the density as a function of the natural logarithm of x as discrete points. For comparison, the solid line shows the function

$$Q(x) = \ln(x)^{1+c},$$

which was first shown by Erdos to give a lower bound on Q(x). Here c is an undetermined constant greater than zero. The solid line is drawn with c= 1/3.

Summary

We have shown an algorithm to quickly calculate highly composite numbers and have used it to extend the list of known highly composite numbers from the 100, previously published, up to 1000.


Q(x)
1000
800
600
400
200
0
0
25
50
75
100
125
150
175
ln(x)

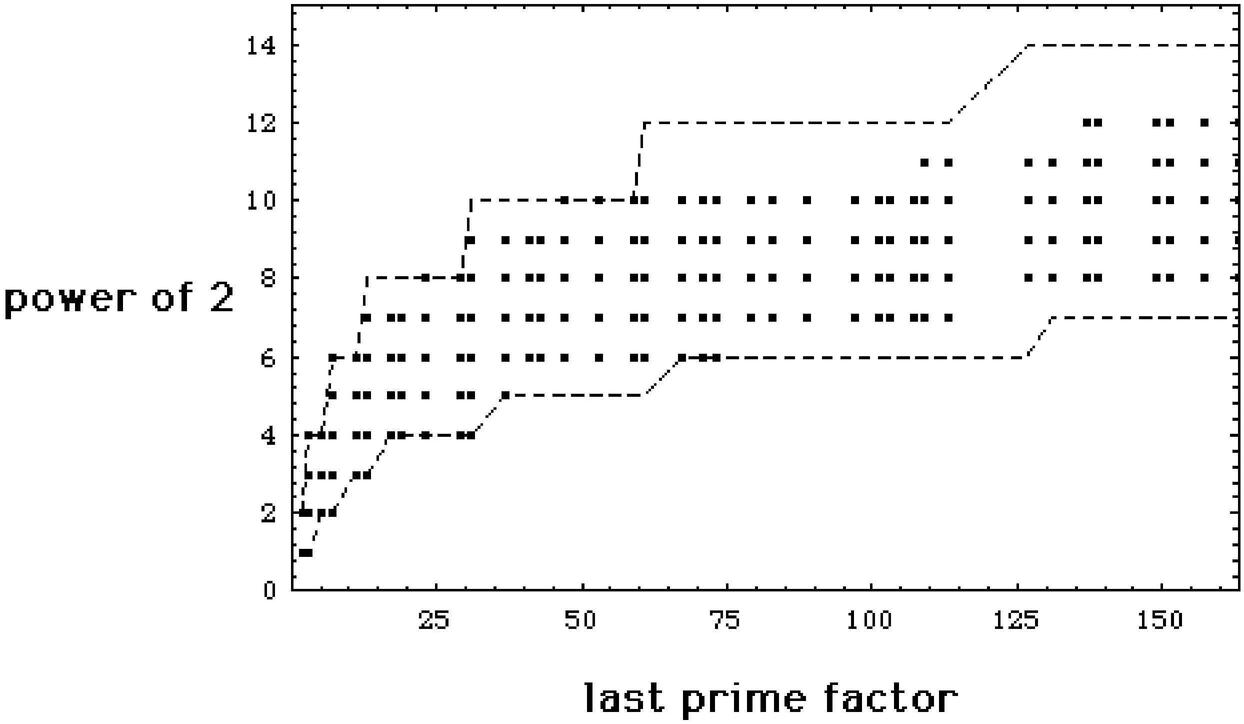
Appendix I.

Correct bounds f and g

The values of f and g to be used for a larger range of desired highly composite numbers can determined as follows. Ramanujan has shown that there is a connection between the power of two, a2 and the largest prime factor, p in a highly composite number:

$$\left[\frac{\log(p)}{\log(2)}\right] \le a_2 \le 2\left[\frac{\log(p_+)}{\log(2)}\right],$$

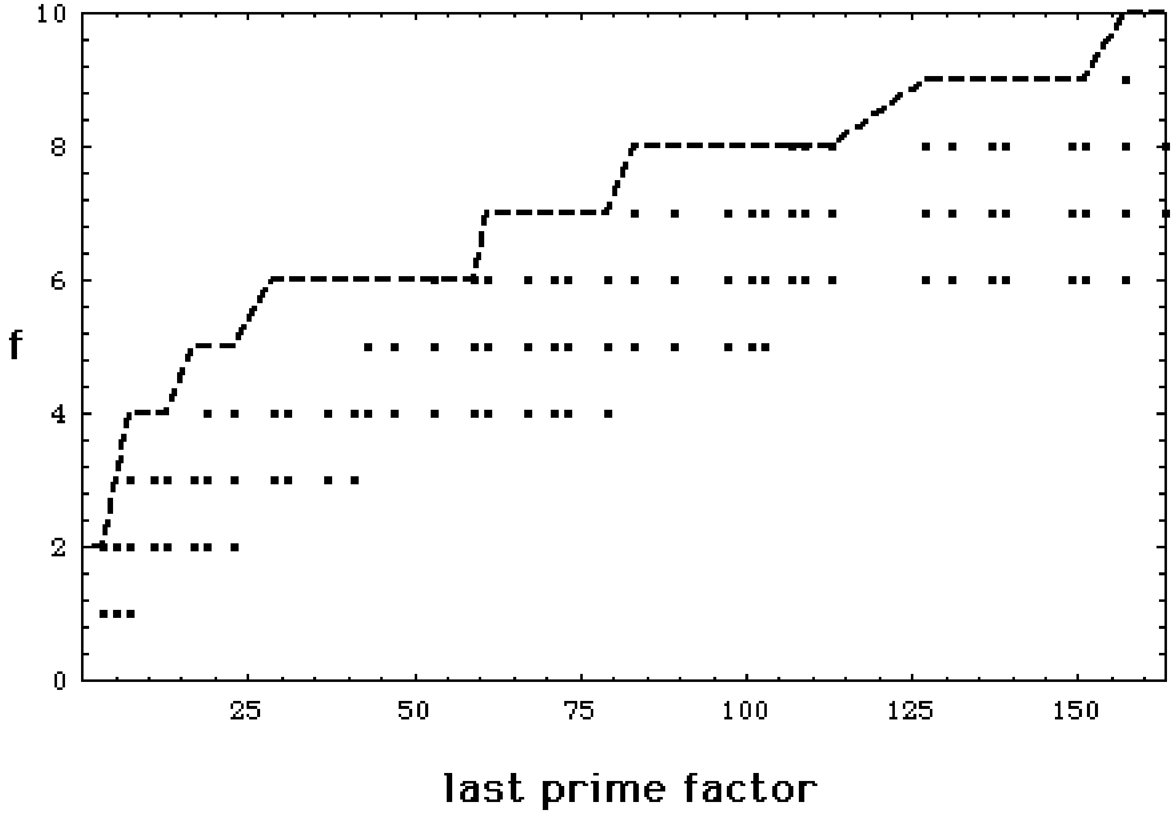
where [] stands for the integral part. Thus, for our example of the 85th highly composite number, p=19 and pp=23 gives a2 between 4 and 8, compared to the actual value of 6. When p=23 and pp=29 the formula also gives a2 between 4 and 8. Therefore g = 8 is good enough to certainly get the next higher highly composite number.



The number, f, of factors in v that is large enough to cover all of the possibilities involves a somewhat more obscure function.

$$\Lambda(\lambda, p) = [1/(2^{\log\lambda/\log p} - 1)]$$

Hereλ is a prime less than p, and, as before, p is the largest prime factor of hc(n). For values of λ greater than some critical value Λ equals unity. The critical value of l therefore determines the size of f, the number of prime exponents that is not one. This is shown in fig. 1 as dashed line. The number actually found in the sequence of the first 1000 highly composite numbers is shown in Fig. 1.



References

1. "Collected Papers of S. Ramanujan", Ed. by G.H. Hardy, PV. Seshu Aiyar and B. M. Wilson, Chelsea Publishing Co. , NY, NY.
2. P. Erdos, J. London Math. Soc. 19, 130-133 (1944).