Journal of Universal Computer Science, vol. 9, no. 10 (2003), 1196-1203
submitted: 6/6/03, accepted: 29/9/03, appeared: 28/10/03 © J.UCS

What Is the Value of *Taxicab*(6)?

Cristian S. Calude
(Department of Computer Science, University of Auckland, New Zealand
Email: cristian@cs.auckland.ac.nz)

Elena Calude
(Institute of Information Sciences, Massey University at Albany, New Zealand
Email: e.calude@massey.ac.nz)

Michael J. Dinneen
(Department of Computer Science, University of Auckland, New Zealand
Email: mjd@cs.auckland.ac.nz)

Abstract: For almost 350 years it was known that 1729 is the smallest integer which can be expressed as the sum of two positive cubes in two different ways. Motivated by a famous story involving Hardy and Ramanujan, a class of numbers called Taxicab Numbers has been defined: $Taxicab(k, j, n)$ is the smallest number which can be expressed as the sum of j kth powers in n different ways. So, $Taxicab(3, 2, 2) = 1729; Taxicab(4, 2, 2) = 635318657$. Computing Taxicab Numbers is challenging and interesting, both from mathematical and programming points of view.

The exact value of $Taxicab(6) = Taxicab(3, 2, 6)$ is not known; however, recent results announced by Rathbun [R2002] show that $Taxicab(6)$ is in the interval $[10^{18}, 24153319581254312065344]$. In this note we show that with probability greater than 99%, $Taxicab(6) = 24153319581254312065344$.

Key Words: Hardy-Ramanujan Number, Taxicab Number, sampling

Category: G.3

1 Is 1729 a Dull Number?

Srinivasa Ramanujan was one of India's greatest, largely self-taught, mathematical geniuses. In January 1913 Ramanujan, after seeing the book *Orders of Infinity*, wrote to its author, the distinguished number theorist G. H. Hardy. Hardy and Littlewod studied the long list of unproved theorems which Ramanujan enclosed with his letter and sent him the letter beginning with the famous paragraph:

> I was exceedingly interested by your letter and by the theorems which you state. You will however understand that, before I can judge properly of the value of what you have done, it is essential that I should see proofs of some of your assertions. Your results seem to me to fall into roughly three classes:

> (1) there are a number of results that are already known, or easily deducible from known theorems;
> (2) there are results which, so far as I know, are new and interesting, but interesting rather from their curiosity and apparent difficulty than their importance;
> (3) there are results which appear to be new and important ...

In 1914 Ramanujan arrived in England and for a couple of years Hardy and Ramanjuan had a collaboration which led to important results. The following is a famous story told by Hardy, repeated word for word in various sources. Ramanjuan became ill in 1917 and at the age of 33 lay dying in Putney hospital. Hardy on a visit to his colleague, moved and at a loss for words, could only say "I came in a taxi 1729, that's a pretty dull number." Ramanujan's immediate rejoinder was "Oh no Hardy. 1729 is the smallest integer which can be expressed in two different ways as the sum of two cubes."[1] This episode prompted Littlewood to say that "every positive integer was one of [Ramanujans'] personal friends".

The number 1729 has since become known as the Hardy-Ramanujan Number, even though this feature of 1729 was known more than 300 years before Ramanujan (more precisely, by Bernard Frénicle de Bessy in 1657, cf. [BB1993]).

Is 1729 really a dull number? Not at all. First, 1729 is a Carmichael Number, i.e., a pseudoprime relative to every base. Also note after [D] that, beginning at the 1729th decimal digit of the transcental number e, the next ten successive digits of e are 0719425863, in the first appearance of all ten digits in a row without repititions.

Finally, are there any dull natural numbers? We can prove that there are none. Indeed, if there are dull numbers, then we can divide all numbers into two sets – interesting and dull. We then pick up the smallest dull number. Since it is the smallest uninteresting number it becomes, *ipso facto*, an interesting number. We must therefore remove it from the dull set and place it in the other set. But now there will be another smallest dull number. Repeating this process will make any dull number interesting.

2 How Many Taxicab Numbers Are Known?

The smallest number expressible as the sum of two cubes in n different ways is called the Taxicab Number $Taxicab(n)$, that is $Taxicab(n) = Taxicab(3, 2, n)$. Hardy and Wright [HW1954] (Theorem 412) have proven that the $Taxicab(n)$

[1] More accurately, "1729 is the smallest integer which can be expressed as the sum of two positive cubes in two different ways."

exists for every positive integer n, but the proof is of little use in finding the number.

Trivially, $Taxicab(1) = 2 = 1^3 + 1^3$. The next number, $Taxicab(2) = 1729 = 1^3 + 12^3 = 9^3 + 10^3$, is the Hardy-Ramanujan Number; see Butcher [B1998] for all solutions of the equation $a^3 + b^3 = A^3 + B^3$. In 1957, Leech [L1957] computed

$$Taxicab(3) = 87539319 = 167^3 + 436^3 = 228^3 + 423^3 = 255^3 + 414^3,$$

in 1991 Rosenstiel, Dardis, and Rosenstiel [RDR1991] (see also Butler's program [B2001]) showed that

$$\begin{aligned} Taxicab(4) = 6963472309248 &= 2421^3 + 19083^3 \\ &= 5436^3 + 18948^3 \\ &= 10200^3 + 18072^3 \\ &= 13322^3 + 16630^3, \end{aligned}$$

and in 1997 Wilson [W1999] discovered the fifth Taxicab Number,

$$\begin{aligned} Taxicab(5) = 48988659276962496 &= 38787^3 + 365757^3 \\ &= 107839^3 + 362753^3 \\ &= 205292^3 + 342952^3 \\ &= 221424^3 + 336588^3 \\ &= 231518^3 + 331954^3. \end{aligned}$$

In 1998 Bernstein (see [B2000]) discovered that 391909274215699968 is a six-way sum of cubes and showed that $Taxicab(6) \geq 10^{18}$. In 2002 Rathbun [R2002] has found a smaller six-way sum of cubes:

$$\begin{aligned} Taxicab(6) \leq 24153319581254312065344 &= 28906206^3 + 582162^3 \\ &= 28894803^3 + 3064173^3 \\ &= 28657487^3 + 8519281^3 \\ &= 27093208^3 + 16218068^3 \\ &= 26590452^3 + 17492496^3 \\ &= 26224366^3 + 18289922^3. \end{aligned}$$

To appreciate the gap between 10^{18} and 24153319581254312065344 note that the last number is approximately $2.4 \cdot 10^{22}$. In May 2003, Gascoigne [G2003] verified that $Taxicab(6) > 6.8 \cdot 10^{19}$.

3 A Probabilistic Approach to the Computation of $Taxicab(6)$

There are various approaches to the computation of $Taxicab$, see [S, P2002, S2003, T]. The main idea is to use an efficient codification and a "computational structure" that supports insertion of new elements and removal of the smallest element (somethimes referred to as "priority queues"). For example, Bernstein [B2000] codes the pairs as

$$a^3 + 2b^3 + 3c^3 = 4d^3 \tag{1}$$

and computes all positive integers $a, b, c, d \leq H$ satisfying (1). The sorting is done in time/space $H^{2+o(1)}$.

None of these approaches may directly work for the calculation of $Taxicab(6)$. So, in what follows we are going to use a sampling approach (see [C1977]; for an application to automata theory see [CCCDN2001]). The result will not be exact, but the error will be less than 1%.

Here is the method. Given a finite population of size N, we will estimate 3 proportions associated with 3 binary random variables, $P_i = P\left(X_i = 1\right), i = 1, 2, 3$, using a pseudo-random sample of size n. Each estimate should be correct within $\pm c\%$ in the sense that, if the sample shows p_i to be P_i, the percentage for the whole population is "sure" to lie between $p_i - c\%$ and $p_i + c\%$ (with "accuracy within $c\%$"). As we cannot guarantee an accuracy within $c\%$, we accept a probability α (0.0027 in our case) of getting "an unlucky sample" (which is in error by more than the desired $c\%$).

The process of random generation of the sample is equivalent with sampling with replacement (generated units are independent, repetitions are possible), according to a Binomial scheme. If $(u_1, \ldots, u_n)$ are generated items, we denote by m_i the number of items for which $X_i(u) = 1$, $i = 1, 2, 3$. Then, the estimates of P_i are $p_i = \frac{m_i}{n}, i = 1, 2, 3$. For a large value of n ($n > 100$), one can use the normal approximation for p_i, that is, p_i is approximately normal distributed $N\left(P_i, \frac{P_i(1-P_i)}{n}\right)$.

In order to estimate the value of n, we start from the simultaneous conditions

$$\Pr\left(\mid p_i - P_i \mid \geq z_{1-\frac{\alpha}{2}} \sqrt{Var\left(p_i\right)}\right) = \alpha, i = 1, 2, 3,$$

where $z_{1-\frac{\alpha}{2}}$ is the $(1 - \frac{\alpha}{2})$–quantile of the $N\left(0, 1\right)$ distribution, and $z_{0.99865} \approx 3$ (see, for instance, [H1965], Table II).

Accordingly, we will have

$$z_{1-\frac{\alpha}{2}} \sqrt{\frac{P_i\left(1 - P_i\right)}{n}} = c, i = 1, 2, 3,$$

hence, the sample size is

$$n = \max\left\{ \frac{z^2_{1-\frac{\alpha}{2}} P_i\,(1-P_i)}{c^2}, i = 1, 2, 3 \right\}.$$

The value of n depends on the unknown proportions P_i. As we don't have any knowledge regarding P_i, we will choose the "critical" value $P = 50\%$ (which maximizes the product $P(1-P)$). Hence, the "safest" estimation of the sample size n is

$$\hat{n} = \frac{z^2_{1-\frac{\alpha}{2}} \cdot 2,500}{c^2}.$$

The parameters α and c are independent; their values determine the sample size. In fact, a simple "reverse engineering" will determine a pair (α, c) such that the size of the sample is reasonable small, but the resulting accuracy is significant. For our level of significance $\alpha = 0.0027$, we obtain $z_{1-\frac{\alpha}{2}} = 3$. For an accuracy within $c = 5\%$ we get $\hat{n} = 900$ (small, less significant) and for $c = 6.04^{1/2} \approx 2.45\%$, $\hat{n} = 5,625$ (small, still less significant); for $c = 1\%$, $\hat{n} = 22,500$ (reasonable small, significant), which is the size of the sample investigated by our program.

4 The Program

Our program reads the sample random numbers (generated in a standard way with Mathematica) and tests them; it either finds a smaller six-way sum of cubes (in which case it improves the current bound, but doesn't necessarily obtain the exact value of $Taxicab(6)$) or gives a strong plausibility argument against the possibility of finding a smaller six-way sum of cubes. *The second alternatives happens in our case.* The program (written for this present research, not as a package application), listed in Appendix, Figure 1, uses version 4.1.2 of GMP (GNU Multiple Precision) library, [GNU].

To save on computation time we first precompute all of the possible cubes up to the maximum input value. These are stored in the vector `v(N)` in increasing order so that we can decide if a number is a cube root in $\lg N$ time (see `binary_search` conditional). On input `str` we subtract each possible cube in `v(N)` from the integer representation `currentSum` and count how many of those differences are also cubes.

Theoretically, in terms of N (the cube root of our input upper bound) the program runs in time $O(N \cdot n \cdot \log N)$ for a single input number of size n bits. So, when $n = \lg_2(N^3)$, the largest input size, we can replace N with $2^{\frac{n}{3}}$ to see that we have an exponential-time $O(2^{\frac{4n}{3}} \cdot n)$ algorithm. However, note that this bound is more pessimistic than the worst-case scenario since not all iterations of our `binary_search` call are carried out.

The memory size of the program is *only* dependant on the variable N, not the number of cases (which depends on c and α). For our computation the program used 800M of main memory and ran for about 10 days on a 1GHz linux machine. We run the program on a sample of 22,500 random numbers in the interval $[10^{18}, 24153319581254312065344)$ and *we found no number satisfying the required condition, hence with probability greater than* 99%, $Taxicab(6) = 24153319581254312065344$.

5 Generalisations of Taxicab Numbers

We can define Taxicab Numbers for higher powers than three as well as for more representations. In general, $Taxicab(k, j, n)$ is the smallest number which can be expressed as the sum of j kth powers in n different ways. For example, $Taxicab(4, 2, 2)$ is the smallest number that is a sum of two fourth powers in two different ways. Euler has shown that

$$Taxicab(4, 2, 2) = 635318657 = 59^4 + 158^4 = 133^4 + 134^4,$$

but no example is known for 3-way sums or more; for fifth powers no example of a 2-way sum is known, Schneider [S2003].

The following Taxicab Numbers can be easily computed:

$$\begin{aligned}
Taxicab(2, 3, 2) &= 62 = 1^2 + 5^2 + 6^2 = 2^2 + 3^2 + 7^2,\\
Taxicab(3, 3, 2) &= 1009 = 1^3 + 2^3 + 10^3 = 24^3 + 6^3 + 9^3,\\
Taxicab(4, 3, 2) &= 6578 = 1^4 + 2^4 + 9^4 = 3^4 + 7^4 + 8^4,\\
Taxicab(5, 3, 2) &= 1375298099 = 3^5 + 54^5 + 62^5 = 24^5 + 28^5 + 67^5,\\
Taxicab(6, 3, 2) &= 160426514 = 3^6 + 19^6 + 22^6 = 10^6 + 15^6 + 23^6.
\end{aligned}$$

When generalizing to higher powers it is not known if n-way sums for higher powers exist, cf. Schneider [S2003].

Other generalisations are obtained by allowing both positive and negative terms. For more information see [S, P2002, S2003, T].

Acknowledgement
The authors wish to thank all four referees for their comments and suggestions.

References

[BB1993] B. C. Berndt, S. Bhargava. Ramanujan–for lowbrows, *Am. Math. Monthly* 100 (1993), 645–656.

[B2000] D. J. Bernstein. Enumerating solutions to $p(a) + q(b) = r(c) + s(d)$, *Mathematics of Computation* 70, 233 (2000), 389–394.

[B1998] J. Butcher. Hardy's taxi, $x^2+3y^2 = p$ and Michael Lennon , *NZMS Newsletter* 76 (1999). `http://www.math.auckland.ac.nz/~butcher/miniature/`.

[B2001] B. Butler. C Program for Ramanujan Quadruples, `http://www.durangobill.com/Rama4.html`, 30 May 2001.

[CCCDN2001] C. S. Calude, Elena Calude, Terry Chiu, Monica Dumitrescu, R. Nicolescu. Testing computational complementarity for Mermin automata, *J. Multi Valued Logic*, 6 (2001), 47–65.

[C1977] W. G. Cochran. *Sampling Techniques*, John Wiley, New York, 1977 (third edition).

[G2003] S. Gascoigne, `http://euler.free.fr/taxicab.htm`, May 2003.

[H1965] A. Hald. *Statistical Tables and Formulas*, John Wiley, New York, 1965.

[HW1954] G. H. Hardy, E. M. Wright. *An Introduction to the Theory of Numbers*, Oxford University Press, London, 1954, 3rd edition.

[L1957] J. Leech. Some solutions of Diophantine equations, *Proc. Cambridge Phil. Soc.* 53 (1957), 778–780.

[P2002] I. Peterson. Taxicab Numbers, *Science News Online*, `http://www.sciencenews.org/20020727/mathtrek.asp`.

[R2002] R. L. Rathbun. Sixth Taxicab Number?, `http://listserv.nodak.edu/scripts/wa.exe?A2=ind0207&L=nmbrthry&P=R530`, July 16, 2002.

[RDR1991] E. Rosenstiel, J. A. Dardis, C. R. Rosenstiel. The four least solutions in distinct positive integers of the Diophantine equation $s = x^3 + y^3 = z^3 + w^3 = u^3 + v^3 = m^3 + n^3$, *Bull. Inst. Math. Appl.* 27 (1991), 155–157.

[S2003] W. Schneider. Taxicab Numbers, `http://www.wschnei.de/number-theory/taxicab-numbers.html`, 3 February 2003.

[W1999] D. W. Wilson. The fifth Taxicab Number is 48 988 659 276 962 496, *J. Integer Seq.* 2 (1999), Article 99.1.9, 1, `http://www.research.att.com/~njas/sequences/JIS/wilson10.html`.

[S] N. J. A. Sloane. Sequence A011541, *On-Line Encyclopedia of Integer Sequences*, `http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A011541`.

[T] The Taxicab Problem, `http://euler.free.fr/taxicab.htm`.

[D] The Dullness of 1729, `http://www.mathpages.com/home/kmath028.htm`.

[GNU] The GNU GMP library, `http://www.swox.com/gmp/`.

APPENDIX

```
/*
 * Program to check for whether a set of integers can be expressed in NR
 * different ways as a sum of two positive integer cubes.  20 May 2003
 */
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include <gmpxx.h>

using namespace std;

const unsigned long N = 28906206;  // cube root of upper bound on input
const unsigned long NR = 6;        // our target taxicab count

int main(){
   mpz_class t1, t2;
   vector<mpz_class> v(N);
   for (unsigned long i=0; i<N; ++i)
                                  {// create the vector of cubic values
      t1 = i+1; t2 = t1*t1; v[i] = t2*t1;
   }
   string str;                        // input number as string
   mpz_class currentSum;              // input number as MP integer
   mpz_class N_a3;                    // test cube
   unsigned long a;                   // test cube root

   while (true){
      cin >> str; if (str.length()==1) break;
      currentSum=str;
      cout << "processing: " << currentSum << endl;

      int cnt=0;
      for (a=0; a<N-1; a++){
         N_a3 = currentSum-v[a];
         if (N_a3 < v[a]) break;

        if ( binary_search(v.begin(), v.end(), N_a3) ){
           cnt++;
           if (cnt > maxcnt) { maxcnt = cnt; cerr << "maxcnt=" <<
           cnt <<endl; }
           vector<mpz_class>::iterator VI = lower_bound(v.begin(),
           v.end(), N_a3);
           cout << "a=" << (a+1) << ' ' << "b=" << (VI-v.begin())+1
           <<endl;
        }
        if (cnt == NR){ cout << "Found taxicab(NR)!" << endl; return 1;}
      }
   }
   return 0;  // end of processing
}
```

Figure 1: A simple C++ program that tests for Taxicab numbers.