ENumerating Solutions to \( p(a) + q(b) = r(c) + s(d) \)

Daniel J. Bernstein

Abstract. Let \( p, q, r, s \) be polynomials with integer coefficients. This paper presents a fast method, using very little temporary storage, to find all small integers \((a, b, c, d)\) satisfying \( p(a) + q(b) = r(c) + s(d) \). Numerical results include all small solutions to \( a^4 + 2b^3 + 3c^3 = 4d^3 \); all small solutions to \( a^4 + b^4 = c^4 + d^4 \); and the smallest positive integer that can be written in 5 ways as a sum of two coprime cubes.

1. Introduction

Let \( H \) be a positive integer. How can one find all positive integers \( a, b, c, d \leq H \) satisfying \( a^3 + 2b^3 + 3c^3 = 4d^3 \)?

The following method is standard. Sort the set \( \{(a^3 + 2b^3, a, b) : a, b \leq H\} \) into increasing order in the first component. Similarly sort \( \{(4d^3 - 3c^3, c, d) : c, d \leq H\} \). Now merge the sorted lists, looking for collisions. The sorting takes time \( H^{1+o(1)} \) and space \( H^{2+o(1)} \).

It does not seem to be widely known that one can save a factor of \( H \) in space. Section 3 explains how to enumerate \( \{(a^3 + 2b^3, a, b)\} \) and \( \{(4d^3 - 3c^3, c, d)\} \) in order, using \( O(H^2) \) heap operations on two heaps of size \( H \). Heaps are reviewed in section 2. The remaining sections of this paper give several numerical examples. See http://pobox.com/~djb/sortedsums.html for a UNIX implementation of most of the algorithms discussed here.

A standard improvement is to split the range of \( a^3 + 2b^3 \) and \( 4d^3 - 3c^3 \) into several (0-adic or \( p \)-adic) intervals. For example, one can separately consider each possibility for \( 4d^3 - 3c^3 \mod 7 \), and skip pairs \((a, b)\) with \( a^3 + 2b^3 \mod 7 \in \{2, 5\} \).

Notes. Lander and Parkin in [11] enumerated solutions to \( a^4 + b^4 = c^4 + d^4 \) using time \( H^{3+o(1)} \) and space \( H^{1+o(1)} \).

Ekl in [2] pointed out that the time of the Lander-Parkin method could be reduced to \( H^{2+o(1)} \). I made the same observation independently in April 1997, when Yuri Tschinkel asked me about the example described in section 4 below. David W. Wilson made the same observation independently in October 1997, for the example described in section 5 below. The difference between my method, Ekl’s method, and the Lander-Parkin method is the difference between a heap, a balanced tree, and an unstructured array.

Received by the editor July 10, 1998 and, in revised form, January 4, 1999.

2000 Mathematics Subject Classification. Primary 11Y50; Secondary 11D25, 11D41, 11P05, 11Y16.

The author was supported by the National Science Foundation under grant DMS–9600083.

2. Heaps

A heap is a sequence \( x_1, x_2, \ldots, x_n \) satisfying \( x_{\lfloor k/2 \rfloor} \leq x_k \) for \( 2 \leq k \leq n \); i.e., \( x_1 \leq x_2, x_1 \leq x_3, x_2 \leq x_4, x_2 \leq x_5, x_3 \leq x_6, x_3 \leq x_7, \) etc.

The smallest element of a heap \( x_1, x_2, \ldots, x_n \) is \( x_1 \). Given \( y, x_2, \ldots, x_n \) into a new heap by the following algorithm. First set \( j \leftarrow 1 \). Then perform the following steps repeatedly: set \( k \leftarrow 2j \); stop if \( k > n \); set \( k \leftarrow k + 1 \) if \( k < n \) and \( x_{k+1} < x_k \); stop if \( y \leq x_k \); exchange \( y \), which is now in the \( j \)th position, with \( x_k \); set \( j \leftarrow k \). The total number of operations here is \( O(\log n) \).

In particular, using \( O(\log n) \) operations, one can permute \( x_n, x_{n-1}, \ldots, x_1 \) into a new heap. By a similar algorithm, also using \( O(\log n) \) operations, one can permute \( x_1, x_2, \ldots, x_n, y \) into a new heap.

Notes. Heaps were published by Williams in [22]. Floyd in [5] pointed out an algorithm using \( O(n) \) operations to permute any sequence of length \( n \) into a new heap.


There are many other data structures that support insertion of new elements and removal of the smallest element. Any such structure is called a priority queue. Examples include leftist trees, as discussed in [3] section 5.2.3; loser selection trees, as discussed in [3] section 5.4.1; balanced trees, as discussed in [3] section 6.2.3; and B-trees, as discussed in [3] section 6.2.4. See also [10] page 152. The reader can replace the heap in section 3 with any priority queue. Beware, however, that some “fast” priority queues are several times bigger and slower than heaps; see, for example, section 10 below.

3. Enumerating sums

Fix \( p, q \in \mathbb{Z}[x] \). This section explains how to print \( \{(p(a) + q(b), a, b) : a, b \leq H\} \) in increasing order in the first component, using space \( H^{1+o(1)} \).

First build a table of \( \{(p(a), a) : a \leq H\} \). Sort the table into increasing order in the first component; say \( p(a_1) \leq p(a_2) \leq \cdots \).

Next build a heap containing \( \{(p(a_1) + q(b), 1, b) : b \leq H\} \). Perform the following operations repeatedly until the heap is empty:

1. Find and remove the smallest element \((y, n, b)\) in the heap.
2. Print \((y, a_n, b)\); by induction \( y = p(a_n) + q(b) \) at this point.
3. Insert \((p(a_{n+1}) - p(a_n) + y, n + 1, b)\) into the heap if \( a_{n+1} \) exists.

Step 1 and step 3 can be combined into a single heap operation.

This algorithm takes time \( H^{1+o(1)} \) for initializations, plus \( H^{o(1)} \) for each of the \( H^2 \) outputs, for a total of \( H^{2+o(1)} \). There are at most \( H \) elements in the heap at any moment.
Refinements. One can easily save half the heap operations if \( p = q \): start with 
\( \{(p(a_n) + p(a_n), n, a_n)\} \); print \((y, b, a_n)\) along with \((y, a_n, b)\) if \(a_n \neq b\).

One can speed up the manipulation of \( y \), and in some cases save space, by storing
\( p(a_2) - p(a_1), p(a_3) - p(a_2), \ldots \) instead of \( p(a_2), p(a_3), \ldots \).

One need not bother building tables of \( n \mapsto a_n \) and \( n \mapsto p(a_n) \) if \( p \) is a sufficiently dull function.

Generalizations. Given functions \( p, q, r, s \) from finite sets \( A, B, C, D \) to an ordered group, one can enumerate \( \{(a, b, c, d) \in A \times B \times C \times D : p(a) + q(b) = r(c) + s(d)\} \) by the same algorithm. For example, one can enumerate small solutions \((a, b, c, d)\) to \( a^3 + 2b^3 = 3c^3 + 4d^3 \) with \( a, b, c, d \in \mathbb{Z}[w]/(w^2 + w + 1) \), using lexicographic order on \( \mathbb{Z}[w]/(w^2 + w + 1) \). See section 10 for another example.

One can restrict attention to a subset of \( A \times B \), simply by skipping to the next suitable \( a \) for each \( b \). See sections 9 and 10 for examples.

There are many functions that are not of the form \( a, b \mapsto p(a) + q(b) \) but that are nevertheless amenable to sorted enumeration. For example, one can apply the method here to any function \( f \) such that \( a \mapsto f(a, b) \) is monotone for each \( b \). See section 6 for an example.

4. Example: \( a^3 + b^3 = c^3 + d^3 \)

There are 12137664 solutions \((a, b, c, d)\) to \( a^3 + b^3 = c^3 + d^3 > 0 \) with \( a \neq c, a \neq d, -10^5 \leq a, b, c, d \leq 10^5 \), and \( a\mathbb{Z} + b\mathbb{Z} + c\mathbb{Z} + d\mathbb{Z} = \mathbb{Z} \). In other words, there are 12137664 rational points of height at most \( 10^5 \) on the surface \( x^3 + y^3 + z^3 = 1 \) away from the lines on the surface.

This computation took \( 1.4 \cdot 10^{13} \) cycles on a Pentium II-350. It takes roughly twice as long to do a similar computation for \( pa^3 + qb^3 = pc^3 + qd^3 \); roughly three times as long for \( pa^3 + pb^3 = rc^3 + sd^3 \); and roughly four times as long for \( pa^3 + qb^3 = rc^3 + sd^3 \).

Notes. Peyre and Tschinkel have checked some of my numerical results and some of their theoretical computations against the best available conjecture. See [16]. Heath-Brown in [8] had previously enumerated solutions to \( a^3 + b^3 = c^3 + 2d^3 \) and \( a^3 + b^3 = c^3 + 3d^3 \) with \(-10^5 \leq a, b, c \leq 10^5 \) by a cubic-time method.

In some cases one can save time by using [8, Theorem 1].

5. Example: Many Equal Sums of Two Positive Cubes

The smallest integer that can be written in \( k \) ways as a sum of two cubes of positive integers is 1729 for \( k = 2 \); 87539319 for \( k = 3 \); 6963472309248 for \( k = 4 \); and 48988659276962496 for \( k = 5 \). There are no 6-way integers below \( 10^{18} \). (There are two other 5-way integers below \( 10^{18} \): 391909274215699968 = 8 \cdot 48988659276962496 and 490593422681271000.)

This computation took \( 7.9 \cdot 10^{14} \) cycles on an UltraSPARC II-296.

Notes. The answer for \( k = 3 \) was found by Leech in [14]. The answer for \( k = 4 \) was found by Rosenstiel, Dardis, and Rosenstiel in [17]. The answer for \( k = 5 \) was found by David W. Wilson in 1997 and independently by me in 1998. There is an answer for every \( k \); see [19] for the best known bounds.
6. Example: Many equal sums of two cubes

The smallest positive integer that can be written in \( k \) ways as a sum of two cubes is 91 for \( k = 2 \); 728 for \( k = 3 \); 2741256 for \( k = 4 \); 6017193 for \( k = 5 \); 1412774811 for \( k = 6 \); 11302198488 for \( k = 7 \); and 137513849003496 for \( k = 8 \). There are no 9-way integers below \( 2 \cdot 10^{17} \). (There are 37 other 8-way integers below \( 2 \cdot 10^{17} \).)

This computation took \( 9.2 \cdot 10^{14} \) cycles on an UltraSPARC II-296. To keep the heap small, I enumerated pairs \((a,b)\) with \( a \geq b/2 \) and \( 1 \leq a^3 + (b-a)^3 \leq 2.5 \cdot 10^{17} \), in order of \( a^3 + (b-a)^3 \); these conditions imply \( 1 \leq b \leq 10^6 \).

Notes. The answers for \( k \in \{5, 6, 7\} \) were found by Randall Rathbun, according to [7, page 141]. The answer for \( k = 8 \) appears to be new.

7. Example: Many equal sums of two coprime cubes

The smallest positive integer that can be written in \( k \) ways as a sum of two cubes of coprime integers is 91 for \( k = 2 \); 3367 for \( k = 3 \); 16776487 for \( k = 4 \); and 506433677359393 for \( k = 5 \). Each of these integers is squarefree. There are no 6-way integers below \( 2 \cdot 10^{17} \). (There is one other 5-way integer, namely 137904678696613339.)

I found these results during the computation described in section 6. A separate computation, skipping pairs \((a,b)\) with a common factor, would have been somewhat faster.

Notes. The answer for \( k = 4 \) was found by Rathbun, according to [7, page 141]. The answer for \( k = 5 \) appears to be new.

Silverman proved in [18] that the number of pairs of integers \((a,b)\) satisfying \( a^3 + b^3 = n \) is bounded by a particular function of the rank over \( \mathbb{Q} \) of the elliptic curve \( x^3 + y^3 = n \), if \( n \) is cubefree. It is not known how tight Silverman’s bound is.

Paul Vojta found that 15170835645 can be written in 3 ways as a sum of two cubes of coprime positive integers.

8. Example: \( a^4 + b^4 = c^4 + d^4 \)

There are 516 solutions \((a,b,c,d)\) to \( a^4 + b^4 = c^4 + d^4 \) with \( 0 < b \leq a \), \( 0 < d \leq c \), \( c < a \leq 10^6 \), and \( aZ + bZ + cZ + dZ = Z \). This computation took roughly \( 10^{15} \) cycles on an UltraSPARC II-296.

The fourth power of \( 10^6 \) does not fit into a 64-bit integer. I actually enumerated values of \((a^4 \mod m) + (b^4 \mod m) + (0 \text{ or } m)\) greater than or equal to \( m \), where \( m = 2^{60} - 93 \). Then I checked each collision \( a^4 + b^4 \equiv c^4 + d^4 \mod m \) to see whether \( a^4 + b^4 = c^4 + d^4 \).

Notes. 218 of the 516 solutions were already known: Lander and Parkin in [11] exhaustively found all solutions with \( a^4 + b^4 < 7.885 \cdot 10^{15} \); Lander, Parkin, and Selfridge in [13] exhaustively found all solutions with \( a^4 + b^4 \leq 5.3 \cdot 10^{16} \); Zajta in [23] found many solutions with \( a \leq 10^6 \) by various ad-hoc techniques.
9. Example: \(a^4 + b^4 + c^4 = d^4\)

The only positive solutions \((a, b, c, d)\) to \(a^4 + b^4 + c^4 = d^4\) with \(d \leq 2.1 \cdot 10^7\) and \(a\mathbb{Z} + b\mathbb{Z} + c\mathbb{Z} + d\mathbb{Z} = \mathbb{Z}\) are permutations of the solutions

\[
(95800, 414560, 217519, 422481),
(1390400, 2767624, 673865, 2813001),
(5507880, 8332208, 1705575, 8707481),
(5870000, 11289040, 8282543, 12197457),
(12552200, 14173720, 4479031, 16003017),
(3642840, 7028600, 16281009, 16430513),
(2682440, 18796760, 15365639, 20615673).
\]

This computation took \(4.5 \cdot 10^{15}\) cycles on a Pentium II-350.

I used several \(p\)-adic restrictions here. One can permute \(a, b, c\) so that \(a \in 2\mathbb{Z}\) and \(b \in 10\mathbb{Z}\). Then \(a \in 8\mathbb{Z}, \ b \in 40\mathbb{Z}, \ d - 1 \in 8\mathbb{Z}, \) and \(c \equiv \pm d \pmod{1024}\) by [20, Theorem 1]; also \(d \notin 5\mathbb{Z}\). There are roughly \(H^2/320\) possibilities for \((a, b)\) and \(H^2/10240\) possibilities for \((c, d)\) if \(d \leq H\). I enumerated sums modulo \(2^{40} - 93\) as in section 5.

**Notes.** Euler conjectured that \(a^4 + b^4 + c^4 = d^4\) had no positive integer solutions. Ward in [20] proved that there are no solutions with \(d \leq 10^4\). Lander, Parkin, and Selfridge in [13] proved that there are no solutions with \(d \leq 2.2 \cdot 10^5\). Elkies in [4] proved that there are infinitely many solutions with \(a\mathbb{Z} + b\mathbb{Z} + c\mathbb{Z} + d\mathbb{Z} = \mathbb{Z}\), and exhibited two examples. Elkies commented that the smaller example, with \(d = 20615673\), “seems beyond the range of reasonable exhaustive computer search.” Frye in [6] subsequently found the solutions with \(d = 422481\), and proved that there are no other solutions with \(d \leq 2 \cdot 10^6\). Allan MacLeod subsequently found the solutions with \(d = 2813001\) by Elkies’s method. The solutions with \(d \in \{8707481, 12197457, 16003017, 16430513\}\) appear to be new.

For each \((c, d)\) satisfying various \(p\)-adic restrictions, Ward factored \(d^4 - c^4\) into primes and then found all representations of \(d^4 - c^4\) as a sum of squares; the total time of Ward’s algorithm is \(H^{2+o(1)}\) with modern factoring methods, but the \(o(1)\) is fairly large. Lander, Parkin, Selfridge, and Frye instead enumerated possibilities for \(b\), and checked for each \(b\) whether \(d^4 - c^4 - b^4\) was a fourth power; Frye estimated that his program used about \(H^3/490000\) fourth-power tests to find all solutions with \(d \leq H\).

10. Example: \(a^7 + b^7 + c^7 + d^7 = e^7 + f^7 + g^7 + h^7\)

The five smallest integers that can be written in 2 ways as sums of four positive seventh powers are \(20563641173794800, \ 12191487610289536, \ 263214614245734400, \ 696885239160606459, \ 1560510411417060608\). There are no other examples below \(420^7\).

I began this computation by generating a sorted table of \(\{a^7 + b^7 : a \geq b\}\). Then I enumerated sums \((a^7 + b^7) + (c^7 + d^7)\) in order, skipping inputs \((a, b), (c, d)\) with \(b \leq c\). Searching up to \(155^7\), to verify the smallest example, took \(1.4 \cdot 10^{10}\) cycles (and roughly 340 kilobytes of memory) on an UltraSPARC I-167. Searching up to \(420^7\) took \(1.4 \cdot 10^{12}\) cycles.
Notes. All the examples here were found by Ekl in [2] and [3]. However, Ekl needed \(1.6 \cdot 10^{11}\) cycles on an HP PRISM-50 (and roughly 8900 kilobytes of memory) to find the first example. Presumably the main reason is that the priority queue in [2] and [3] was a balanced tree, whereas the priority queue here is a heap.

References

16. Emmanuel Peyre, Yuri Tschinkel, Tanagawa numbers of diagonal cubic surfaces, numerical evidence, this journal, previous article.
21. Ingo Wegener, Bottom-up-heapsort, a new variant of heapsort, beating, on average, quicksort (if \(n\) is not very small), Theoretical Computer Science 118 (1993), 81–98. MR 94c:68007

Department of Mathematics, Statistics, and Computer Science (M/C 249), The University of Illinois at Chicago, Chicago, IL 60607–7045

E-mail address: djb@pobox.com