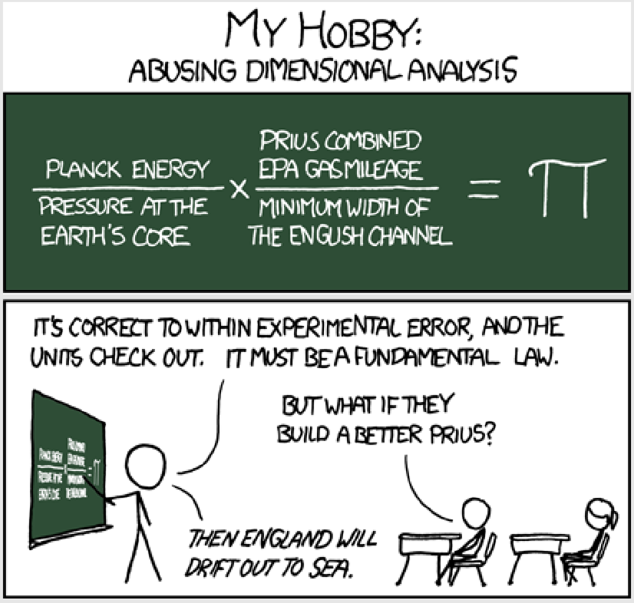
MINING FOR NOBEL

Thore Husfeldt and Rasmus Pagh. Revision b70b075, Thu Feb 16 10:39:45 2012 +0100.



xkcd 687

Deep-looking formulas. On 15 February 2012, around 6 pm, Thore Husfeldt discovered the following amazing formula

$$m_{\mathrm{u}}c^2 m_{\mathrm{d}}c^2 \sum_{k=0}^{\infty} \frac{(-1)^k}{a_k - 1} = \frac{1}{G} \qquad (1)$$

where $m_{\mathrm{u}}c^2$ is the energy equivalent of the atomic mass constant in in MeV and $m_{\mathrm{d}}c^2$ is the energy equivalent of the mass of a deuterium atom in Joule. The next value is Cahen's Constant (roughly 0.643410), defined by

$$C = \sum_{k=0}^{\infty} \frac{(-1)^k}{a_k - 1},$$

where a_i is the ith Sylvester number. Finally, G is who you're gonna call, i.e., the phone number of the Ghostbusters (5552368). Really. In the words of American astronomer Maria Mitchell, "Every formula which expresses a law of nature is a hymn of praise to God." Up to rounding errors, of course.

The formula was found by looking for relationships in a large table of important-looking constants, which is the idea of this exercise. We will look for formulas involving four values A, B, C, D, of the form

$$A \cdot B \cdot C \cdot D = 1\,.$$

Sums are easier to handle than products, so after taking natural logarithms we get the sum problem

$$a + b + c + d = 0\,,$$

where $a = \ln A$, $b = \ln B$, and so on. So that's the task: you get a list of natural logarithms of a few hundred natural constants and have to find sets of four values whose sum is zero. (Of course, the funny values are just window-dressing to make this exercise less boring. It's really just a very simple exercise about four-sums.)

Exhaustive search. The simplest solution is to use exhaustive search using four nested for loops. Go ahead and implement that first. Play around with it by running it on simple input files (that you must generate yourself) of various small sizes. Perform an experimental analysis of the running time, see the README file. Make sure it works correctly by running it on a small file that you have constructed yourself. (For example, with the values 0.11, 0.22, 0.33, and −0.66.) Your programme should output the number of four-sums. To keep things simple we allow repetitions of the same value, so solutions of the form $a + b + a + b = 0$ are fine. Also, let's agree that $a + b + c + d$ and $a + b + d + c$ count as two different sums; this also makes everything a bit easier.

Close to cubic time. Use the idea of the fast three-sum algorithm in [SW, Chapter 1.4] `ThreeSumFast` to write a faster four-sum algorithm. Your running time must be proportional to $N^3 \lg N$. This is the code you need to hand in, call it `MiningNobel.java`.

Close to quadratic time. There is a cute and simple way to solve the four-sum problem in time proportional to $N^2 \lg N$ instead. If you can figure that trick out, implement it and hand in that code instead. Hint: it uses sorting.

Rounding.

I have rounded all values to the nearest power of 2^{-20}. If you handle them as Java Doubles, and don't invent too many fancy things, you should get the same result as me without worrying about about rounding errors. But it's not something we take very seriously in this exercise.

About the input. The file constants.csv contains a list of comma-separated constants (or rather, their natural logarithms). It's been compiled by Rasmus Pagh and Thore Husfeldt from various sources. I can find 528 four-sums in that file, including the amazing formula (1). (One word about files of comma-separated values: your operating system is probably set up to open CSV files in a *spreadsheet programme*. Don't get confused by that. This is not an exercise about spreadsheets.) You have to read the CSV file from standard input and parse it.

input format:

```
tenth Fibonacci prime , 19.887389183044434
answer to everything , 3.737668991088867
ways to leave your lover␣,␣3.912022590637207
```

one constant per line

description *space comma space* *ln of value*