Foursum Report

Alice Cooper and Bob Marley

February 11, 2018

Exhaustive search

[1]

Our program [`Simple.java` / `simple.py`][2] solve the Four-sum problem using four nested loops. The index variables i, j, k, l run from [...][3] Thus, we can bound the number of array accesses by $\sim \frac{1}{10}(\cos N)\binom{N}{5}$.[4]

[1] Complete the report by filling in your names, filling in the parts marked [...] and changing other parts wherever necessary. (For instance, the numbers is the tables are nonsense right now.) Remove the sidenotes in your final hand-in.
[2] Choose one.
[3] Explain. Do you start in 0? One sentence.
[4] Replace by an expression that corresponds to your code.

Experiments

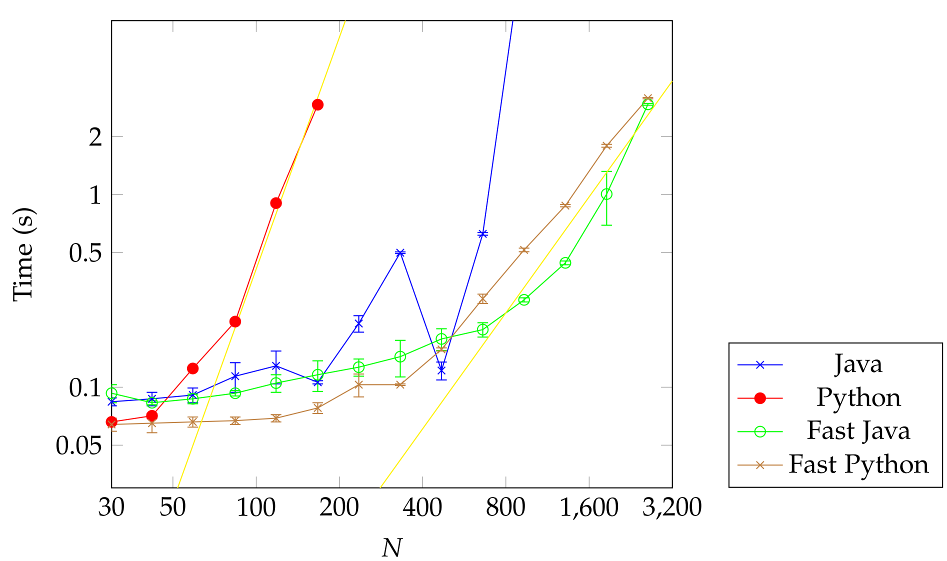
The following table summarises the empirical performance data on the input files in the `data` directory. We have run each file once, and report the minimum, maximum, and average running time over the files for each input size.

	Java			Python		
N	min	max	avg.	min	max	avg.
100	1 sec	5 days	23.5 hours			
[...]						

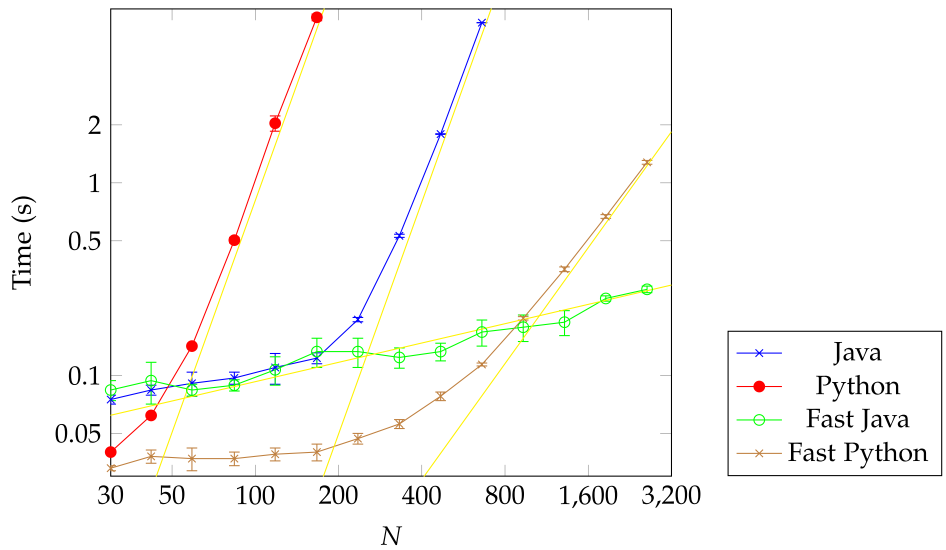
We can plot the maximum running time as a function of input size, as a standard plot and as a log–log plot.[5] In the latter, note that the points fall nicely on a line of slope [...][6].

[5] Draw both graphs, as two separate figures. Use any software you want, or draw it hand.
[6] replace by an integer

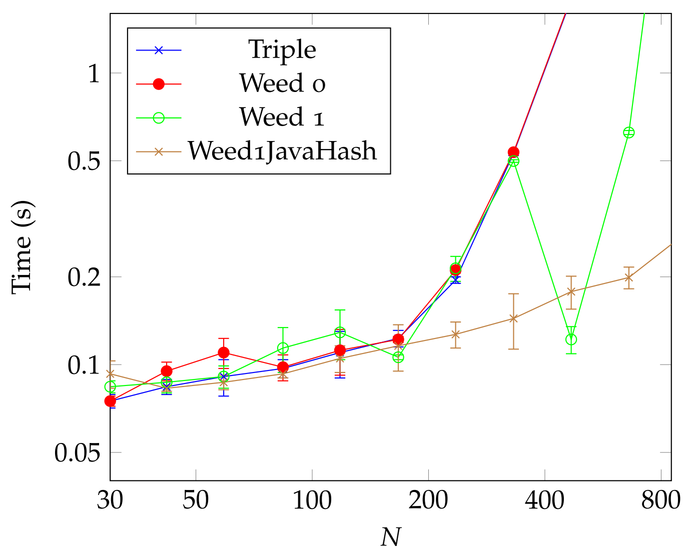
Running time for search, one planted quadruple



Running time for exhaustive search



Running time for simple search



Improvements

Using the binary search-based idea sketeched in [SW, 1.4] for the Three-sum problem, we can improve our running time to $\sim N^3 \log N$. [7]

The following table reports our maximum running times on the test inputs. [$\cdots$]

The following log–log plot shows these values graphically, note that the points are on a line of slope [$\cdots$].

[7] Correct as necessary. If instead you've figured out how to do it even faster, rewrite the whole sentence so as to explain what you do.