Write-up

Hypothesis

The hash set will outperform the ordered set when inserting into a large set.

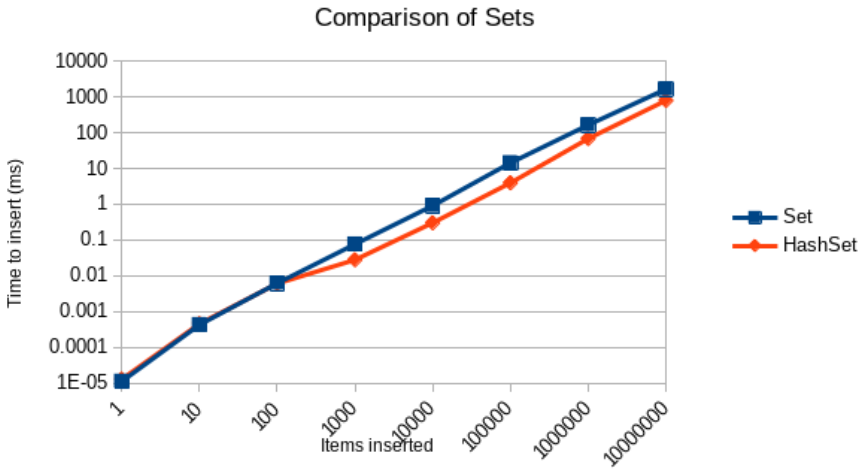
Methods

The code I used to test these algorithms can be found at https://github.com/khuldraeseth/cse431hw4. Clone the repository, move into `p3`, and `stack run -- --output wherever-you-want-the-results.html` to run these tests yourself. Compiles with `-threaded`.

I used sets (`Data.Set.Set` and `Data.HashSet.HashSet`) instead of maps because I saw no reason to do otherwise. I stopped at ten million, before either type of set exceeded three seconds, because of memory and hardware limitations.

Results

See below for Criterion output. The results are summarized in this plot:



Discussion

The hash set pulls away somewhere between 100 and 1000 items, and the ordered set is never able to catch up. What surprised me is how close the ordered sat managed to stay—I would expect the O(n) hash set to outperform the O(n log n) ordered set by more and more, but this never happened. In fact, the ordered set took less time as a proportion of that taken by the hash set to insert ten million items than to insert one thousand.

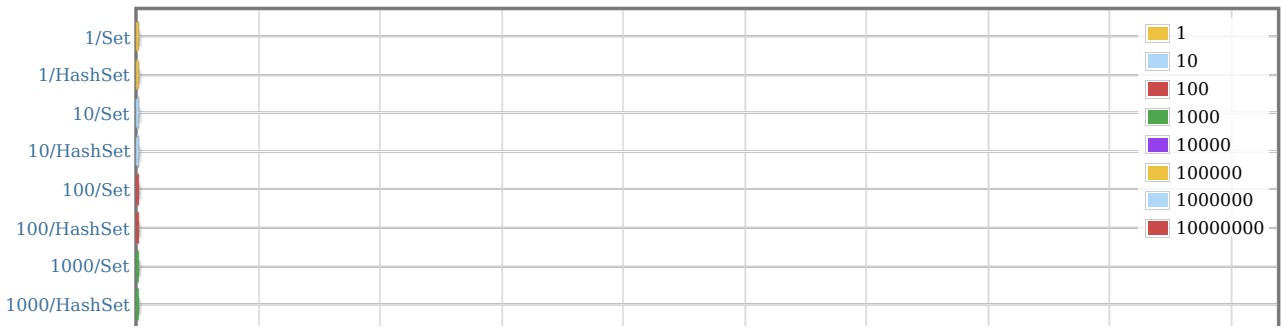
Conclusion

Under the conditions tested, a hash set could be more quickly filled with at least 100 items than an ordered set.

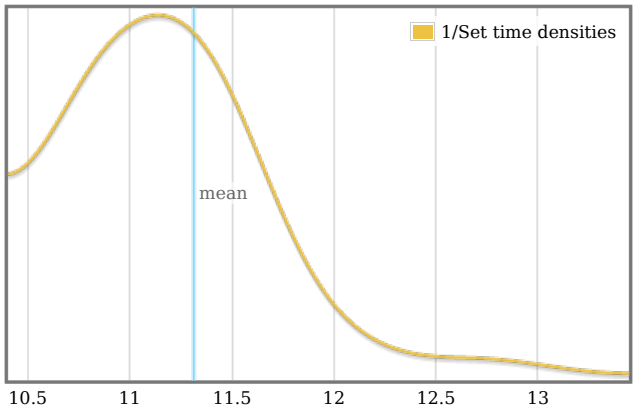
criterion performance measurements

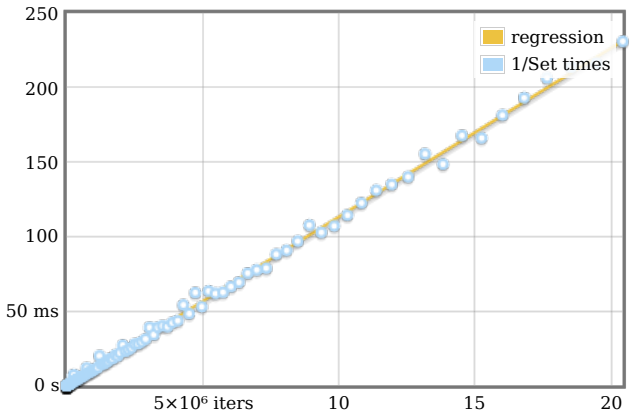
overview

want to understand this report?



1/Set

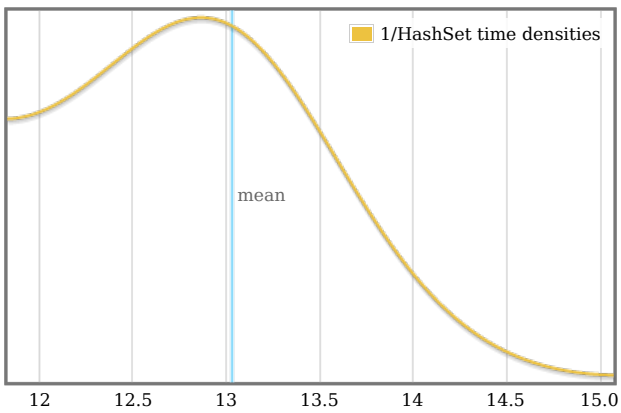


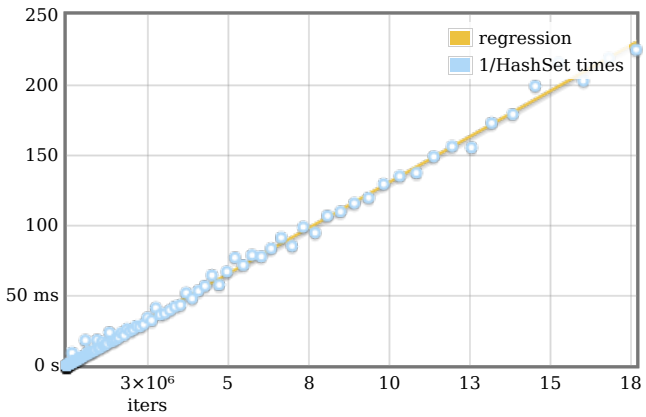


	lower bound	estimate	upper bound
OLS regression	11.2 ns	11.3 ns	11.4 ns
R² goodness-of-fit	0.998	0.999	0.999
Mean execution time	11.2 ns	11.3 ns	11.6 ns
Standard deviation	447 ps	585 ps	888 ps

Outlying measurements have severe (75.1%) effect on estimated standard deviation.

1/HashSet

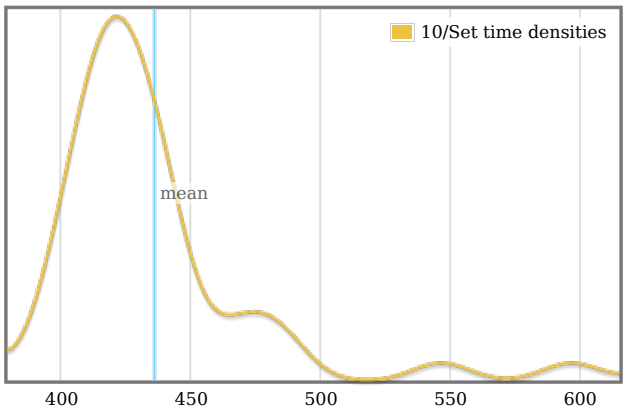


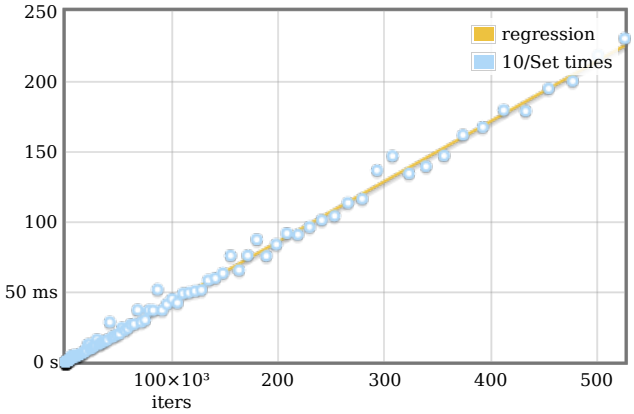


	lower bound	estimate	upper bound
OLS regression	12.8 ns	13.0 ns	13.3 ns
R² goodness-of-fit	0.997	0.998	0.999
Mean execution time	12.8 ns	13.0 ns	13.2 ns
Standard deviation	558 ps	687 ps	846 ps

Outlying measurements have severe (75.9%) effect on estimated standard deviation.

10/Set

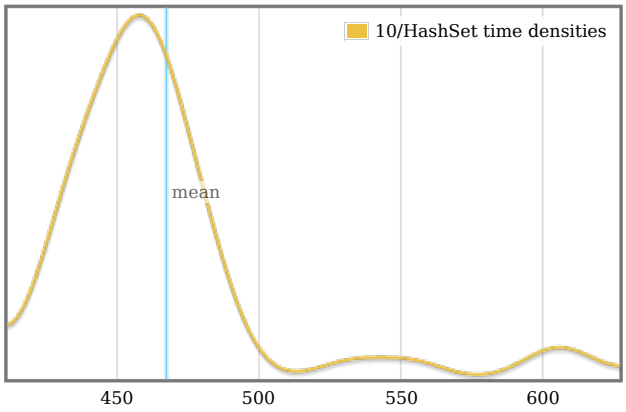


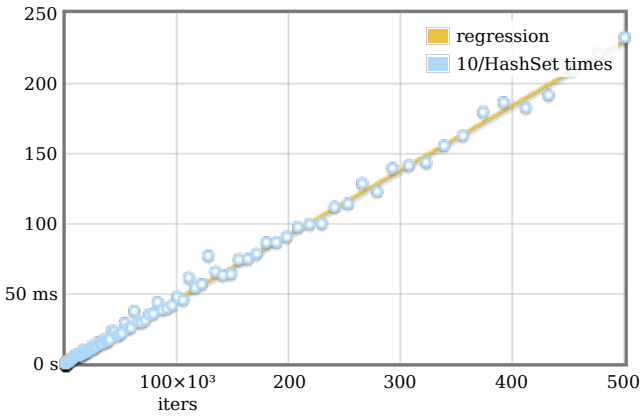


	lower bound	estimate	upper bound
OLS regression	423 ns	429 ns	436 ns
R² goodness-of-fit	0.995	0.997	0.998
Mean execution time	428 ns	436 ns	452 ns
Standard deviation	23.4 ns	38.2 ns	61.5 ns

Outlying measurements have severe (87.0%) effect on estimated standard deviation.

10/HashSet

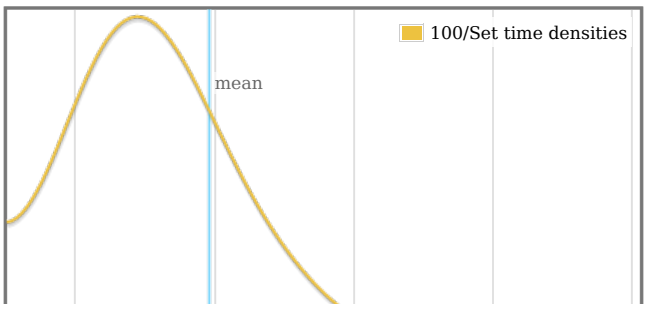


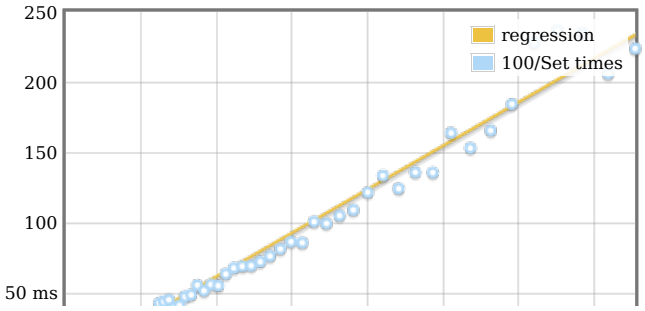


	lower bound	estimate	upper bound
OLS regression	454 ns	460 ns	466 ns
R² goodness-of-fit	0.995	0.998	0.999
Mean execution time	458 ns	468 ns	481 ns
Standard deviation	26.4 ns	39.5 ns	60.2 ns

Outlying measurements have severe (86.1%) effect on estimated standard deviation.

100/Set

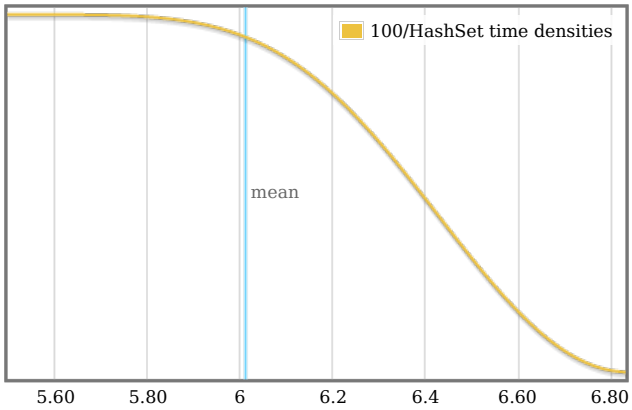


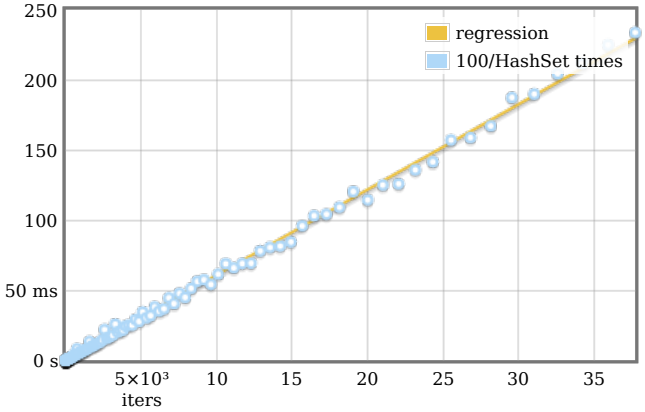


	lower bound	estimate	upper bound
OLS regression	5.87 µs	6.18 µs	6.49 µs
R² goodness-of-fit	0.985	0.989	0.998
Mean execution time	5.85 µs	5.98 µs	6.14 µs
Standard deviation	364 ns	487 ns	625 ns

Outlying measurements have severe (81.3%) effect on estimated standard deviation.

100/HashSet

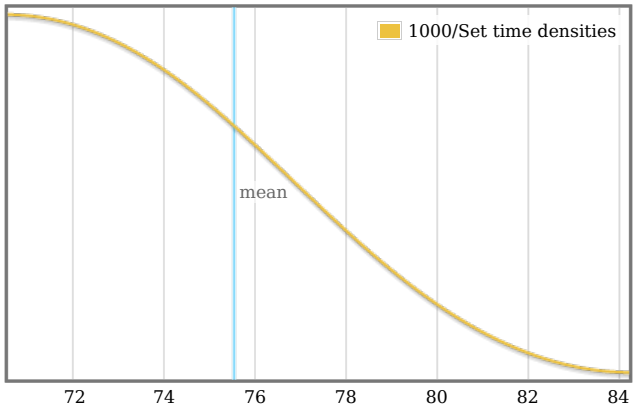


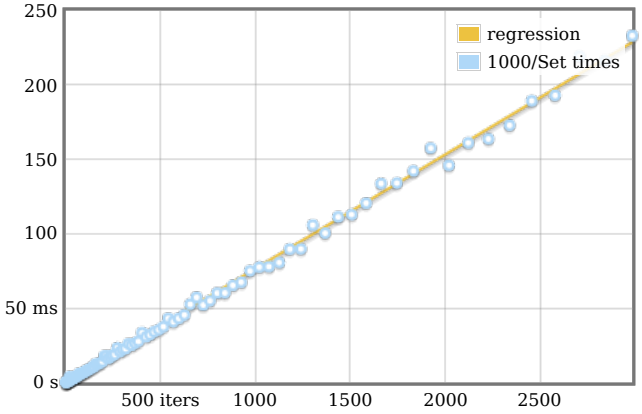


	lower bound	estimate	upper bound
OLS regression	5.98 µs	6.08 µs	6.15 µs
R² goodness-of-fit	0.998	0.998	0.999
Mean execution time	5.93 µs	6.01 µs	6.10 µs
Standard deviation	262 ns	299 ns	355 ns

Outlying measurements have severe (61.8%) effect on estimated standard deviation.

1000/Set

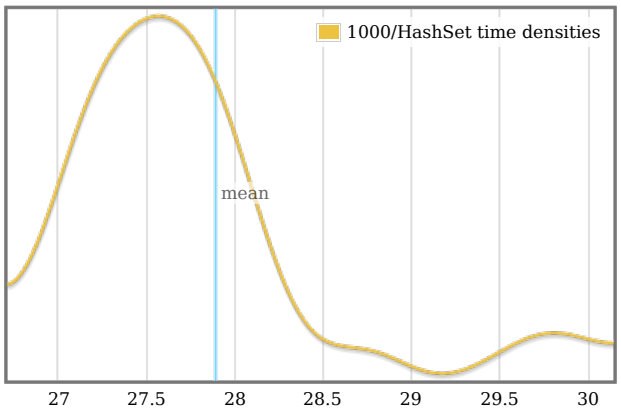


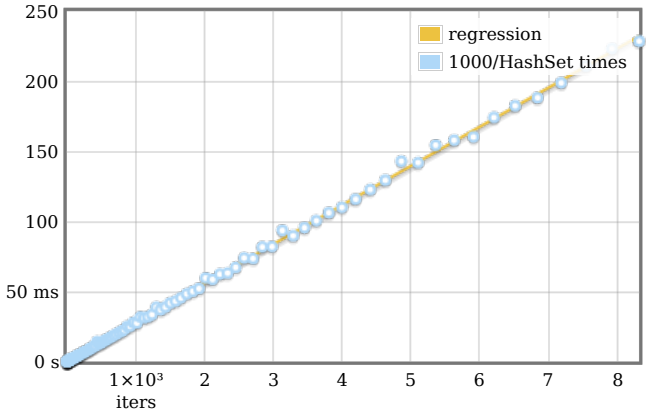


	lower bound	estimate	upper bound
OLS regression	75.2 µs	76.6 µs	77.8 µs
R² goodness-of-fit	0.997	0.998	0.999
Mean execution time	74.4 µs	75.5 µs	76.4 µs
Standard deviation	2.80 µs	3.40 µs	4.01 µs

Outlying measurements have moderate (47.7%) effect on estimated standard deviation.

1000/HashSet

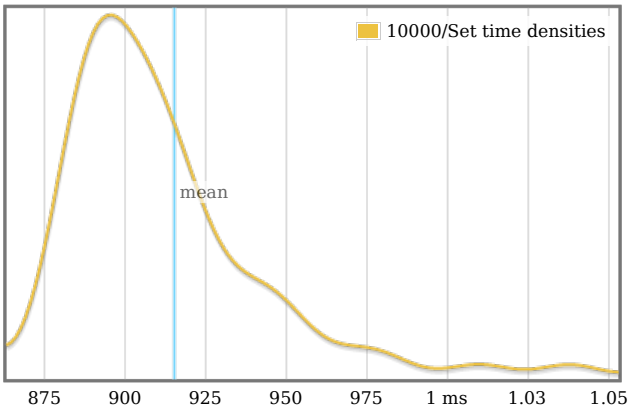


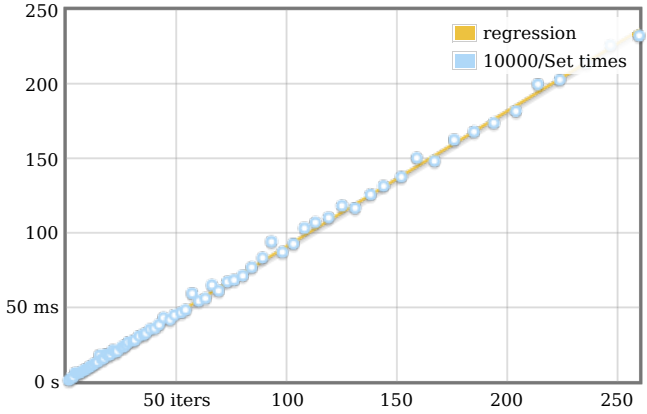


	lower bound	estimate	upper bound
OLS regression	27.7 µs	27.9 µs	28.1 µs
R² goodness-of-fit	0.999	0.999	1.000
Mean execution time	27.7 µs	27.9 µs	28.1 µs
Standard deviation	617 ns	793 ns	992 ns

Outlying measurements have moderate (29.5%) effect on estimated standard deviation.

10000/Set

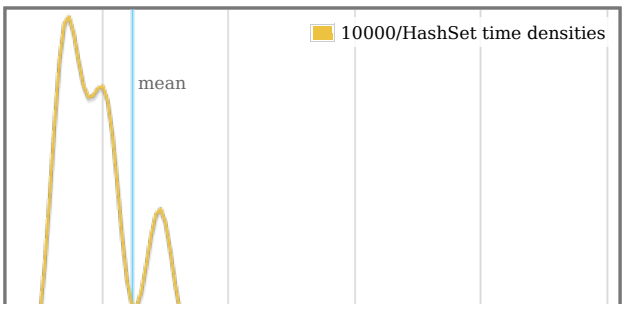


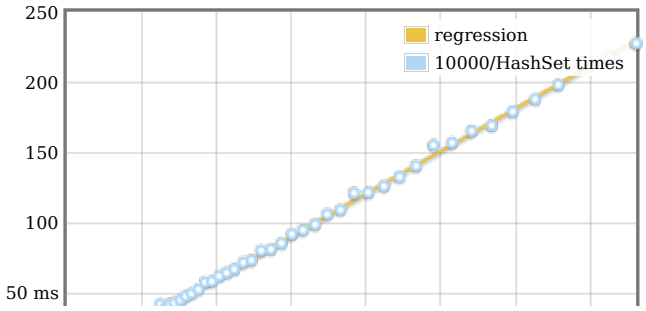


	lower bound	estimate	upper bound
OLS regression	900 µs	909 µs	917 µs
R² goodness-of-fit	0.998	0.999	0.999
Mean execution time	908 µs	916 µs	928 µs
Standard deviation	24.4 µs	34.0 µs	46.5 µs

Outlying measurements have moderate (26.6%) effect on estimated standard deviation.

10000/HashSet

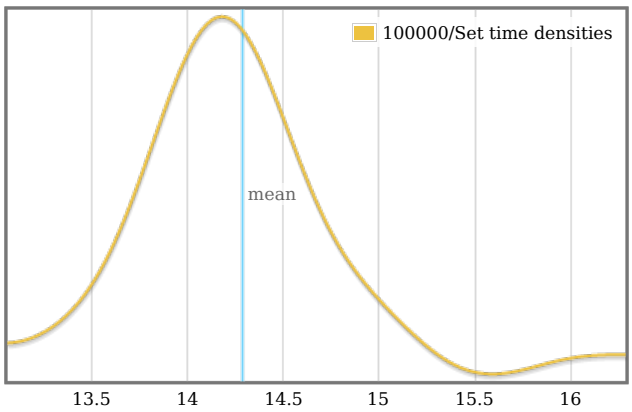


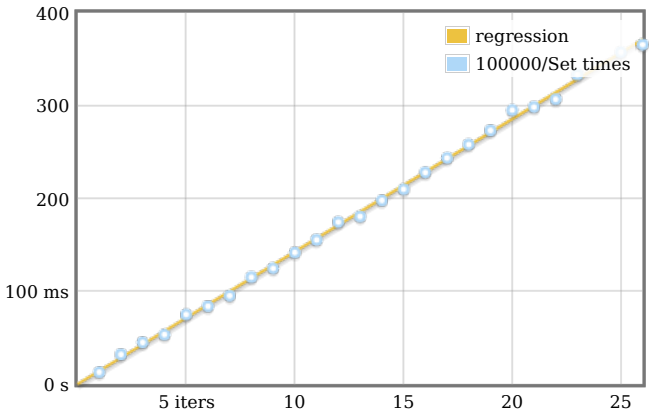


	lower bound	estimate	upper bound
OLS regression	300 µs	302 µs	305 µs
R² goodness-of-fit	0.999	0.999	1.000
Mean execution time	301 µs	302 µs	305 µs
Standard deviation	4.56 µs	7.44 µs	11.9 µs

Outlying measurements have moderate (17.2%) effect on estimated standard deviation.

100000/Set

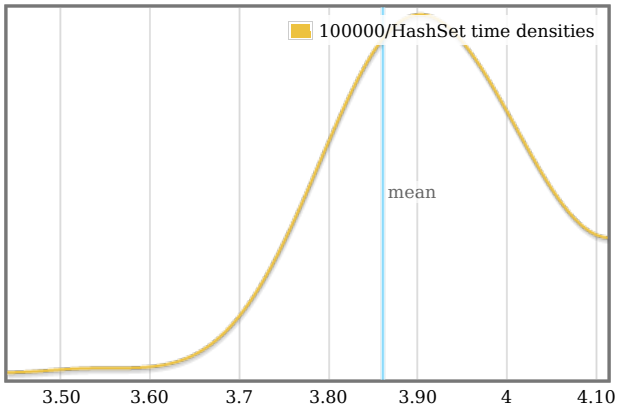


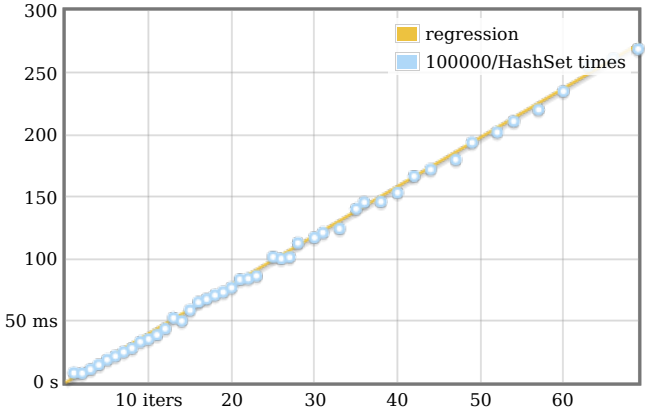


	lower bound	estimate	upper bound
OLS regression	14.1 ms	14.3 ms	14.5 ms
R² goodness-of-fit	0.998	0.999	0.999
Mean execution time	14.1 ms	14.3 ms	14.5 ms
Standard deviation	338 µs	533 µs	826 µs

Outlying measurements have moderate (11.4%) effect on estimated standard deviation.

100000/HashSet

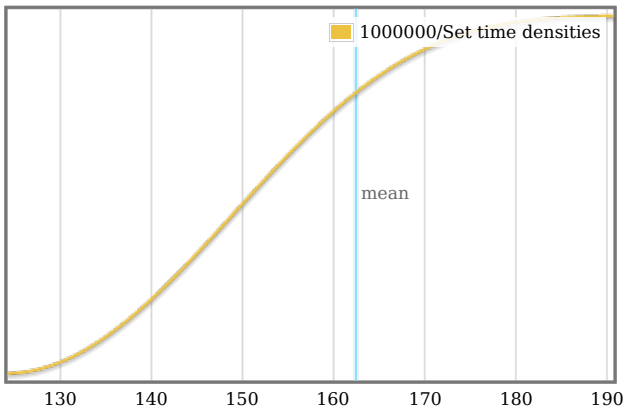


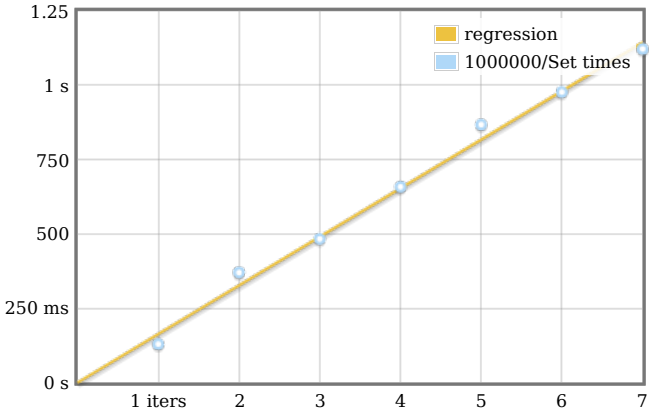


	lower bound	estimate	upper bound
OLS regression	3.90 ms	3.95 ms	4.00 ms
R² goodness-of-fit	0.998	0.999	0.999
Mean execution time	3.81 ms	3.86 ms	3.90 ms
Standard deviation	116 µs	146 µs	188 µs

Outlying measurements have moderate (19.0%) effect on estimated standard deviation.

1000000/Set

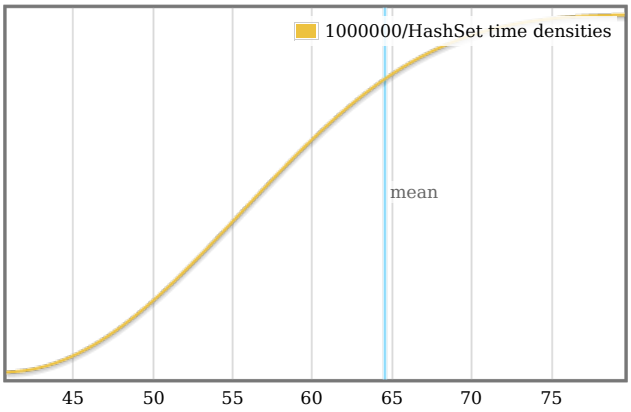


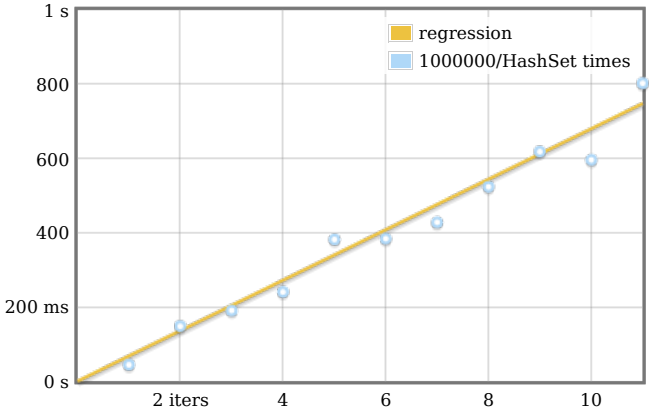


	lower bound	estimate	upper bound
OLS regression	151 ms	163 ms	180 ms
R² goodness-of-fit	0.981	0.992	1.000
Mean execution time	148 ms	162 ms	172 ms
Standard deviation	8.49 ms	16.9 ms	25.3 ms

Outlying measurements have moderate (26.9%) effect on estimated standard deviation.

1000000/HashSet

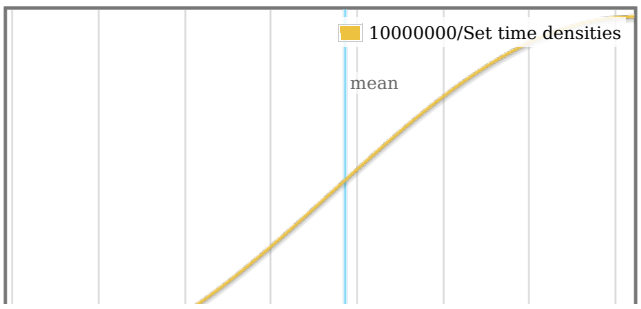


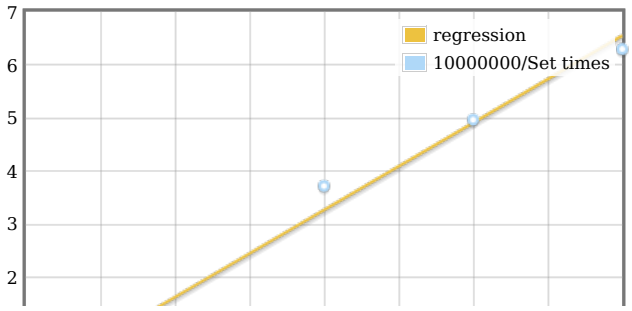


	lower bound	estimate	upper bound
OLS regression	58.6 ms	68.0 ms	75.3 ms
R² goodness-of-fit	0.932	0.971	0.993
Mean execution time	58.2 ms	64.5 ms	68.3 ms
Standard deviation	5.52 ms	8.92 ms	13.6 ms

Outlying measurements have moderate (43.9%) effect on estimated standard deviation.

10000000/Set

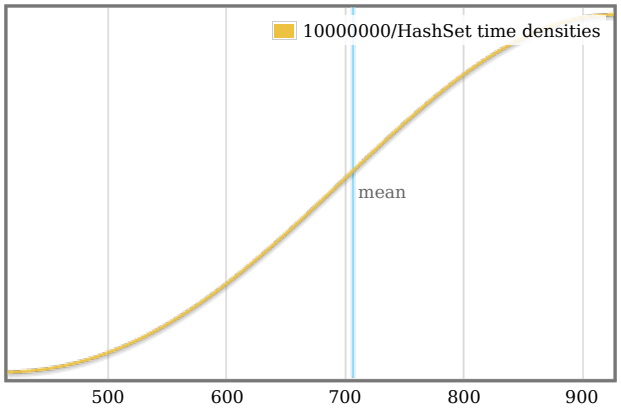


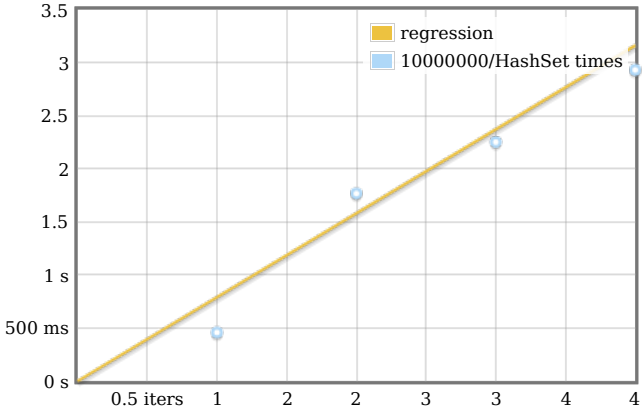


	lower bound	estimate	upper bound
OLS regression	1.25 s	1.64 s	2.47 s
R² goodness-of-fit	0.949	0.970	1.000
Mean execution time	1.33 s	1.59 s	1.74 s
Standard deviation	119 ms	252 ms	351 ms

Outlying measurements have moderate (46.1%) effect on estimated standard deviation.

10000000/HashSet





	lower bound	estimate	upper bound
OLS regression	485 ms	790 ms	1.31 s
R² goodness-of-fit	0.905	0.954	1.000
Mean execution time	527 ms	706 ms	812 ms
Standard deviation	69.9 ms	178 ms	245 ms

Outlying measurements have moderate (48.5%) effect on estimated standard deviation.

understanding this report

In this report, each function benchmarked by criterion is assigned a section of its own. The charts in each section are active; if you hover your mouse over data points and annotations, you will see more details.

- The chart on the left is a kernel density estimate (also known as a KDE) of time measurements. This graphs the probability of any given time measurement occurring. A spike indicates that a measurement of a particular time occurred; its height indicates how often that measurement was repeated.
- The chart on the right is the raw data from which the kernel density estimate is built. The *x* axis indicates the number of loop iterations, while the *y* axis shows measured execution time for the given number of loop iterations. The line behind the values is the linear regression prediction of execution time for a given number of iterations. Ideally, all measurements will be on (or very near) this line.

Under the charts is a small table. The first two rows are the results of a linear regression run on the measurements displayed in the right-hand chart.

- *OLS regression* indicates the time estimated for a single loop iteration using an ordinary least-squares regression model. This number is more accurate than the *mean* estimate below it, as it more effectively eliminates measurement overhead and other constant factors.
- *R² goodness-of-fit* is a measure of how accurately the linear regression model fits the observed measurements. If the measurements are not too noisy, R² should lie between 0.99 and 1, indicating an excellent fit. If the number is below 0.99, something is confounding the accuracy of the linear model.
- *Mean execution time* and *standard deviation* are statistics calculated from execution time divided by number of iterations.

We use a statistical technique called the bootstrap to provide confidence intervals on our estimates. The bootstrap-derived upper and lower bounds on estimates let you see how accurate we believe those estimates to be. (Hover the mouse over the table headers to see the confidence levels.)

A noisy benchmarking environment can cause some or many measurements to fall far from the mean. These outlying measurements can have a significant inflationary effect on the estimate of the standard deviation. We calculate and display an estimate of the extent to which the standard deviation has been inflated by outliers.

colophon

This report was created using the criterion benchmark execution and performance analysis tool.

Criterion is developed and maintained by Bryan O'Sullivan.