

# Final Paper

## An Investigation into Multi-Pivot Quicksort Algorithms

Authors:

Paymahn Moghadasian [umpaymah@myumanitoba.ca]

Joshua Hernandez [umhern23@myumanitoba.ca]

Instructor: Dr. S. Durocher

COMP 4420

Advanced Design and Analysis of Algorithms

Due : April 9, 2014

# Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>1</b>
<b>3</b>	<b>Quicksort</b>	<b>2</b>
3.1	Classic Quicksort . . . . .	2
3.2	Dual Pivot Quicksort . . . . .	2
3.3	Optimal Dual Pivot Quicksort . . . . .	2
3.4	Three Pivot Quicksort . . . . .	2
3.5	Yaroslavskiy Quicksort . . . . .	2
3.6	M Pivot Quicksort . . . . .	2
3.6.1	Testing . . . . .	2
3.7	Summary . . . . .	2
<b>4</b>	<b>Analysis</b>	<b>3</b>
4.1	Data Processing . . . . .	4
4.2	Non-Linear Curve Fit . . . . .	5
4.3	Discussion . . . . .	5
4.4	Summary . . . . .	5
<b>5</b>	<b>Future Work</b>	<b>11</b>
<b>6</b>	<b>Conclusion</b>	<b>11</b>
	<b>References</b>	<b>12</b>
<b>A</b>	<b>GitHub Repository</b>	<b>12</b>

## References

- [1] C. A. R. Hoare, “Quicksort,” *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. 1
- [2] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro, “Multi-pivot quicksort: Theory and experiments,” 2013. 1
- [3] R. Sedgwick and K. Wayne, “Advanced topics in sorting,” 2007. 1
- [4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 1

## 1 Abstract

The quicksort is a thoroughly studied sorting algorithm and is commonly among the first efficient sorts learned by students of computer science. Many variants of the quicksort have been proposed, from the classic quicksort introduced by Tony Hoare in 1961 [1] to Yaroslavskiy’s dual pivot quicksort introduced in 2009 and used by the Java 7 Standard Library [2]. Since Hoare’s first proposal, much research has gone into attempting to minimize the total number of swaps done by the sort, the total number of comparisons done by the sort and minimizing the worst case runtime. We aim to experimentally validate the swap and comparison count of several variants of the quicksort and compare the runtimes and various optimizations. Our results

SUMMARIZE THE RESULTS

## 2 Introduction

Sorting is a fundamental concept of computer science wherein a totally ordered multiset is modified such that the elements of the multiset are rearranged (permuted) in either non-decreasing or non-increasing order. A broad range of applications benefit from sorting from organizing an MP3 library by song title to quickly identifying duplicates in a list to more advanced applications such as load balancing, data compression and computer graphics [3]. It is well known that all comparison based sorting algorithms are lower bound by  $\Omega(n \log n)$  comparisons [4] and quicksort is no exception to this rule. Interestingly, there are non-comparison based sorts such as the counting sort and the radix sort which take advantage of certain properties of the data set and get around the lower bound of comparison base sorts. A summary of space and time complexities can be found in Table 1.

Sort Method	Space	Average Case Time	Worst Case Time
Selection	$O(1)$	$O(n^2)$	$O(n^2)$
Insertion	$O(1)$	$O(n^2)$	$O(n^2)$
Merge	$O(n)$	$O(n \log(n))$	$O(n \log(n))$
Quicksort	$O(1)$	$O(n \log(n))$	$O(n^2)$
Radix	$O(n)$	$O(n * k)$	$O(n * k)$
Counting	$O(m)$	$O(n + m)$	$O(n + m)$

Table 1: Summary of space and time complexities of various sorts where  $n$  represents the number of elements,  $k$  represents the number of digits in the largest value and  $m$  represents the max value to be sorted.

### 3 Quicksort

#### 3.1 Classic Quicksort

#### 3.2 Dual Pivot Quicksort

#### 3.3 Optimal Dual Pivot Quicksort

#### 3.4 Three Pivot Quicksort

#### 3.5 Yaroslavskiy Quicksort

#### 3.6 M Pivot Quicksort

RAWR

##### 3.6.1 Testing

Just to test the subsection code. Test text : Recall from section 3.1 on page 2

#### 3.7 Summary

## References

- [1] C. A. R. Hoare, “Quicksort,” *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. 1
- [2] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro, “Multi-pivot quicksort: Theory and experiments,” 2013. 1

Sort Method	Comparisons
Classic	$2n \log n - 1.51n + O(\log(n))$
Dual Pivot	$2.13n \log n - 2.57n + O(\log(n))$
Optimal Dual Pivot	$1.8n \log n + O(n)$
Three Pivot	$1.846n \log n + O(n)$
Yaroslavskiy	$1.9n \log n - 2.46n + O(\log(n))$
M Pivot	$O(n \log n)$

Table 2: Summary table of theoretical comparisons.

Sort Method	Swaps
Classic	$0.33n \log n - 0.58n + O(\log(n))$
Dual Pivot	$0.8n \log n - 0.3n + O(\log(n))$
Optimal Dual Pivot	$0.33n \log n + O(n)$
Three Pivot	$0.615n \log n + O(n)$
Yaroslavskiy	$0.6n \log n + 0.08n + O(\log(n))$
M Pivot	$O(n \log n)$

Table 3: Summary table of theoretical swaps.

- [3] R. Sedgewick and K. Wayne, “Advanced topics in sorting,” 2007. 1
- [4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 1

## 4 Analysis

So we have established the details of the differences between the different variations of quicksorts. To investigate and compare the algorithms we have implemented and ran experiments. We used uniformly random numbers from 0 to  $10^{12}$ . Then created arrays with randomly generated integers for sizes 4 to 50 million Table 4 shows the variations of quicksort to the arrays generated.

Figure 4 is the legend of all the plots generated. You can see the color and shape combination for each version of quick sort algorithm. These color and shape choices will be consistent throughout the following plots. There are 17 distinct derivatives of quicksort that have been run. Note that for the plots we overlay the function :

$$A \cdot n \log(n) + B \cdot n + C \log(n) \quad (1)$$

The nomenclature for the legend is as follows:

Name of Sort Algorithm	Pivot Selection Methods	Number of Pivots
Classic Quicksort	3	1
Dual Pivot Quicksort	2	2
Heap Optimized M-Pivot Quicksort	1	3,4,5,6
M-Pivot Quicksort	1	3,4,5,6
Optimal Dual Pivot Quicksort	2	2
Three Pivot Quicksort	1	3
Yaroslavskiy Quicksort	1	2

Table 4: List of all the variation of quicksorts executed.

(Algorithm Name, Pivot Selection Method Index, Number of Pivots, is Insertion Sort used).

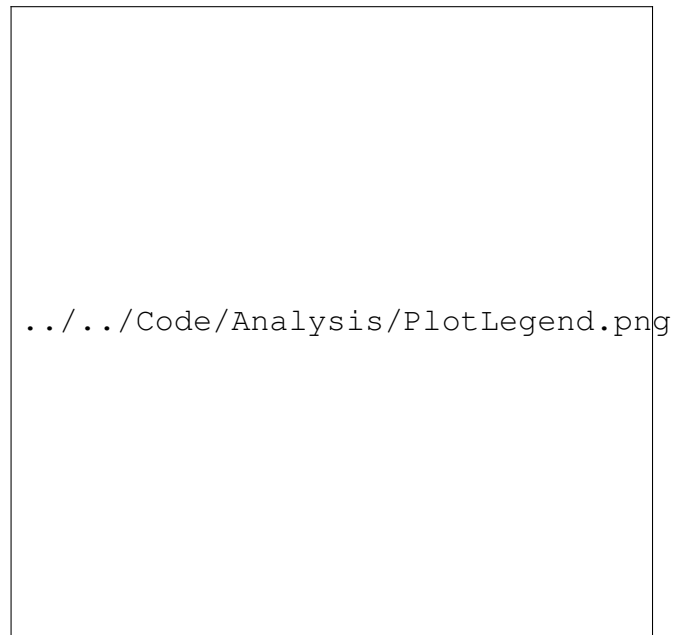


Figure 1: A visual reference for all the quicksort variations and plots.

## 4.1 Data Processing

In processing the data, we take averages of any common data entries. We plot them and have the following results. Which is summarized in Figures 2, 4, 5, and 6.

Sort Method	$A_{\text{comparisons}}$
Classic Quicksort - 1 - 1	$0.02151 \pm 0.00019$
Classic Quicksort - 2 - 1	$0.02135 \pm 0.00017$
Classic Quicksort - 3 - 1	$0.01807 \pm 0.00006$
DualPivot Quicksort - 1 - 2	$0.02014 \pm 0.00018$
DualPivot Quicksort - 2 - 2	$0.01772 \pm 0.00007$
Heap Optimized M-Pivot Quicksort - 1 - 3	$0.02778 \pm 0.00014$
Heap Optimized M-Pivot Quicksort - 1 - 4	$0.02788 \pm 0.00015$
Heap Optimized M-Pivot Quicksort - 1 - 5	$0.02842 \pm 0.00025$
Heap Optimized M-Pivot Quicksort - 1 - 6	$0.02869 \pm 0.00011$
M-Pivot Quicksort - 1 - 3	$0.01970 \pm 0.00009$
M-Pivot Quicksort - 1 - 4	$0.02076 \pm 0.00015$
M-Pivot Quicksort - 1 - 5	$0.02165 \pm 0.00010$
M-Pivot Quicksort - 1 - 6	$0.02386 \pm 0.00014$
Optimal Dual Pivot Quicksort - 1 - 2	$0.01959 \pm 0.00019$
Optimal Dual Pivot Quicksort - 2 - 2	$0.01744 \pm 0.00007$
Three Pivot Quicksort - 1 - 3	$0.02587 \pm 0.00009$
Yaroslavskiy Quicksort - 1 - 2	$0.01796 \pm 0.00010$

Table 5: Summary table coefficients of the non-linear fit for the parameter  $A$  on the comparison data.

## 4.2 Non-Linear Curve Fit

With the function parameters  $A, B$  and  $C$  where found using a curve fitter in the SciPy module in python. As you will see in the following plots. The function are good fits. We have not done analysis to see if we have over fit our data. Although from the analysis done in many of the papers, the functions have that form. Also for the small scale plots. The curve fitted function may not be seen, as a result, we just connect the data points and only show the fitted curve in the large scale plots.

## 4.3 Discussion

Preliminary observations on Figure 2 show that the optimal dual pivot quicksort had the lowest number of comparisons. This is as expected since it was designed to minimize the number of comparisons.

## 4.4 Summary

Sort Method	$B_{\text{comparisons}}$
Classic Quicksort - 1 - 1	$-0.05025 \pm 0.00469$
Classic Quicksort - 2 - 1	$-0.04634 \pm 0.00428$
Classic Quicksort - 3 - 1	$-0.02126 \pm 0.00163$
DualPivot Quicksort - 1 - 2	$-0.03650 \pm 0.00465$
DualPivot Quicksort - 2 - 2	$-0.01045 \pm 0.00183$
Heap Optimized M-Pivot Quicksort - 1 - 3	$-0.05055 \pm 0.00355$
Heap Optimized M-Pivot Quicksort - 1 - 4	$-0.05152 \pm 0.00368$
Heap Optimized M-Pivot Quicksort - 1 - 5	$-0.04639 \pm 0.00626$
Heap Optimized M-Pivot Quicksort - 1 - 6	$-0.03889 \pm 0.00280$
M-Pivot Quicksort - 1 - 3	$-0.01917 \pm 0.00220$
M-Pivot Quicksort - 1 - 4	$-0.01716 \pm 0.00391$
M-Pivot Quicksort - 1 - 5	$-0.00902 \pm 0.00244$
M-Pivot Quicksort - 1 - 6	$-0.03206 \pm 0.00366$
Optimal Dual Pivot Quicksort - 1 - 2	$-0.03580 \pm 0.00490$
Optimal Dual Pivot Quicksort - 2 - 2	$-0.01266 \pm 0.00165$
Three Pivot Quicksort - 1 - 3	$-0.04282 \pm 0.00232$
Yaroslavskiy Quicksort - 1 - 2	$-0.01636 \pm 0.00251$

Table 6: Summary table coefficients of the non-linear fit for the parameter  $B$  on the comparison data.



Sort Method	$C_{\text{comparisons}}$
Classic Quicksort - 1 - 1	$121.34341 \pm 99.97322$
Classic Quicksort - 2 - 1	$78.33233 \pm 91.31275$
Classic Quicksort - 3 - 1	$22.52742 \pm 34.75034$
DualPivot Quicksort - 1 - 2	$52.22569 \pm 99.17037$
DualPivot Quicksort - 2 - 2	$-53.25104 \pm 39.07336$
Heap Optimized M-Pivot Quicksort - 1 - 3	$94.09226 \pm 75.71998$
Heap Optimized M-Pivot Quicksort - 1 - 4	$124.92512 \pm 78.46872$
Heap Optimized M-Pivot Quicksort - 1 - 5	$56.97527 \pm 133.52044$
Heap Optimized M-Pivot Quicksort - 1 - 6	$29.84293 \pm 59.67899$
M-Pivot Quicksort - 1 - 3	$51.60730 \pm 46.87649$
M-Pivot Quicksort - 1 - 4	$49.35746 \pm 83.31083$
M-Pivot Quicksort - 1 - 5	$4.76647 \pm 52.03322$
M-Pivot Quicksort - 1 - 6	$136.38815 \pm 77.96382$
Optimal Dual Pivot Quicksort - 1 - 2	$56.95127 \pm 104.39452$
Optimal Dual Pivot Quicksort - 2 - 2	$-26.20004 \pm 35.22097$
Three Pivot Quicksort - 1 - 3	$-17.79746 \pm 49.38183$
Yaroslavskiy Quicksort - 1 - 2	$0.73204 \pm 53.59187$

Table 7: Summary table coefficients of the non-linear fit for the parameter  $C$  on the comparison data.

Sort Method	$A_{\text{swap}}$
Classic Quicksort - 1 - 1	$0.01026 \pm 0.00017$
Classic Quicksort - 2 - 1	$0.01095 \pm 0.00016$
Classic Quicksort - 3 - 1	$0.00848 \pm 0.00012$
DualPivot Quicksort - 1 - 2	$0.00629 \pm 0.00010$
DualPivot Quicksort - 2 - 2	$0.00606 \pm 0.00006$
Heap Optimized M-Pivot Quicksort - 1 - 3	$0.01004 \pm 0.00009$
Heap Optimized M-Pivot Quicksort - 1 - 4	$0.00898 \pm 0.00004$
Heap Optimized M-Pivot Quicksort - 1 - 5	$0.00809 \pm 0.00004$
Heap Optimized M-Pivot Quicksort - 1 - 6	$0.00759 \pm 0.00005$
M-Pivot Quicksort - 1 - 3	$0.00672 \pm 0.00006$
M-Pivot Quicksort - 1 - 4	$0.00605 \pm 0.00003$
M-Pivot Quicksort - 1 - 5	$0.00535 \pm 0.00003$
M-Pivot Quicksort - 1 - 6	$0.00513 \pm 0.00003$
Optimal Dual Pivot Quicksort - 1 - 2	$0.00629 \pm 0.00010$
Optimal Dual Pivot Quicksort - 2 - 2	$0.00606 \pm 0.00006$
Three Pivot Quicksort - 1 - 3	$0.00635 \pm 0.00006$
Yaroslavskiy Quicksort - 1 - 2	$0.00586 \pm 0.00005$

Table 8: Summary table coefficients of the non-linear fit for the parameter  $A$  on the swap data.

Sort Method	$B_{\text{swap}}$
Classic Quicksort - 1 - 1	$-0.00132 \pm 0.00442$
Classic Quicksort - 2 - 1	$-0.01411 \pm 0.00417$
Classic Quicksort - 3 - 1	$0.01903 \pm 0.00298$
DualPivot Quicksort - 1 - 2	$0.00824 \pm 0.00264$
DualPivot Quicksort - 2 - 2	$0.01080 \pm 0.00153$
Heap Optimized M-Pivot Quicksort - 1 - 3	$0.00774 \pm 0.00233$
Heap Optimized M-Pivot Quicksort - 1 - 4	$0.01111 \pm 0.00097$
Heap Optimized M-Pivot Quicksort - 1 - 5	$0.01881 \pm 0.00113$
Heap Optimized M-Pivot Quicksort - 1 - 6	$0.02363 \pm 0.00129$
M-Pivot Quicksort - 1 - 3	$0.01160 \pm 0.00156$
M-Pivot Quicksort - 1 - 4	$0.01890 \pm 0.00069$
M-Pivot Quicksort - 1 - 5	$0.03171 \pm 0.00065$
M-Pivot Quicksort - 1 - 6	$0.03601 \pm 0.00077$
Optimal Dual Pivot Quicksort - 1 - 2	$0.00824 \pm 0.00264$
Optimal Dual Pivot Quicksort - 2 - 2	$0.01080 \pm 0.00153$
Three Pivot Quicksort - 1 - 3	$0.01030 \pm 0.00153$
Yaroslavskiy Quicksort - 1 - 2	$0.01592 \pm 0.00127$

Table 9: Summary table coefficients of the non-linear fit for the parameter  $B$  on the swap data.

Sort Method	$C_{\text{swap}}$
Classic Quicksort - 1 - 1	$-36.52075 \pm 94.28550$
Classic Quicksort - 2 - 1	$102.34036 \pm 88.94077$
Classic Quicksort - 3 - 1	$-103.21696 \pm 63.57056$
DualPivot Quicksort - 1 - 2	$-38.02577 \pm 56.38166$
DualPivot Quicksort - 2 - 2	$42.34411 \pm 32.52471$
Heap Optimized M-Pivot Quicksort - 1 - 3	$-53.81114 \pm 49.78070$
Heap Optimized M-Pivot Quicksort - 1 - 4	$-3.24127 \pm 20.75454$
Heap Optimized M-Pivot Quicksort - 1 - 5	$-16.07787 \pm 24.00667$
Heap Optimized M-Pivot Quicksort - 1 - 6	$-20.99394 \pm 27.47669$
M-Pivot Quicksort - 1 - 3	$41.76450 \pm 33.26955$
M-Pivot Quicksort - 1 - 4	$16.90493 \pm 14.70501$
M-Pivot Quicksort - 1 - 5	$-17.36329 \pm 13.95680$
M-Pivot Quicksort - 1 - 6	$-5.54593 \pm 16.46902$
Optimal Dual Pivot Quicksort - 1 - 2	$-38.02577 \pm 56.38166$
Optimal Dual Pivot Quicksort - 2 - 2	$42.34411 \pm 32.52471$
Three Pivot Quicksort - 1 - 3	$14.68107 \pm 32.72260$
Yaroslavskiy Quicksort - 1 - 2	$6.27071 \pm 27.13874$

Table 10: Summary table coefficients of the non-linear fit for the parameter  $C$  on the swap data.

## References

- [1] C. A. R. Hoare, “Quicksort,” *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. 1
- [2] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro, “Multi-pivot quicksort: Theory and experiments,” 2013. 1
- [3] R. Sedgwick and K. Wayne, “Advanced topics in sorting,” 2007. 1
- [4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 1

## 5 Future Work

Read the comments in the .tex file

## 6 Conclusion

We did things and stuff!

## References

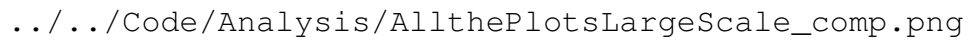
- [1] C. A. R. Hoare, “Quicksort,” *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. 1
- [2] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro, “Multi-pivot quicksort: Theory and experiments,” 2013. 1
- [3] R. Sedgewick and K. Wayne, “Advanced topics in sorting,” 2007. 1
- [4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 1

## References

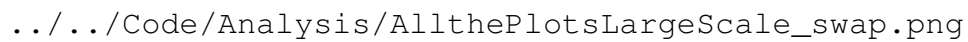
- [1] C. A. R. Hoare, “Quicksort,” *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. 1
- [2] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro, “Multi-pivot quicksort: Theory and experiments,” 2013. 1
- [3] R. Sedgewick and K. Wayne, “Advanced topics in sorting,” 2007. 1
- [4] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 1

## A GitHub Repository

Project GitHub Repository Link



```
../../../../Code/Analysis/AllthePlotsLargeScale_comp.png
```



```
../../../../Code/Analysis/AllthePlotsLargeScale_swap.png
```

Figure 2: A plot of the data from all the sorting algorithm against the number of comparisons.




`../../../../Code/Analysis/AllPlotsLargeScalelognvsvy_OVER_nlogn_comp.png`




`../../../../Code/Analysis/AllPlotsLargeScalelognvsvy_OVER_nlogn_swap.png`

Figure 3: A plot of the data from all the sorting algorithm





```
../../../../Code/Analysis/TwoPivotsLargeScale_comp.png
```



```
../../../../Code/Analysis/TwoPivotsLargeScale_swap.png
```

Figure 4: A plot of the data from all the sorting algorithm with two pivots

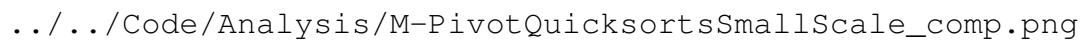


```
../../Code/Analysis/ThreePivotsLargeScale_comp.png
```

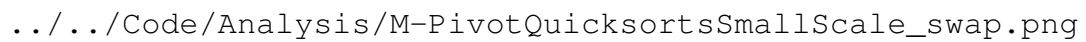


```
../../Code/Analysis/ThreePivotsLargeScale_swap.png
```

Figure 5: A plot of the data from all the sorting algorithm with three pivots



`../../Code/Analysis/M-PivotQuicksortsSmallScale_comp.png`



`../../Code/Analysis/M-PivotQuicksortsSmallScale_swap.png`

Figure 6: Data from all the version of M-Pivot Sort.