All Sorts of Quicksorts!

An investigation into multi-pivot quicksorts

Paymahn Moghadasian Joshua Hernandez

University of Manitoba

April 7, 2014

Introduction

- $O(n \log(n))$ Average Case Run Time

Introduction

- $O(n \log(n))$ Average Case Run Time
- In place algorithm

Introduction

- $O(n \log(n))$ Average Case Run Time
- In place algorithm
- Picking Pivots

Introduction

- $O(n \log(n))$ Average Case Run Time
- In place algorithm
- Picking Pivots
- Partitioning Data

Introduction

- $O(n \log(n))$ Average Case Run Time
- In place algorithm
- Picking Pivots
- Partitioning Data
- Recurse to a smaller sub-array

Introduction

- $O(n \log(n))$ Average Case Run Time
- In place algorithm
- Picking Pivots
- Partitioning Data
- Recurse to a smaller sub-array
- Use Insertion Sort for a small sub-array

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot
  - First Element

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot
  - First Element
  - Last Element

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot
  - First Element
  - Last Element
  - Median of Three

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot
  - First Element
  - Last Element
  - Median of Three
- Simple Partitioning

[Wild and Nebel(2012)]

Classic Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.33n \log n - 0.58n + O(\log(n))$ Swaps
- One Pivot
  - First Element
  - Last Element
  - Median of Three
- Simple Partitioning
- Two Recursive Calls

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots
  - First and Last Element

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots
  - First and Last Element
  - Middle 2 of 5 elements (Evenly Spaced Out)

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots
  - First and Last Element
  - Middle 2 of 5 elements (Evenly Spaced Out)
- Partitions Smalls then Bigs (Middle is automatic)

[Wild and Nebel(2012)]

Dual Pivot Quicksort

- $2.13n \log n - 2.57n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots
  - First and Last Element
  - Middle 2 of 5 elements (Evenly Spaced Out)
- Partitions Smalls then Bigs (Middle is automatic)
- Three Recursive Calls

[Wild and Nebel(2012)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons

[Aumüller and Dietzfelbinger(2013)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons
- $0.33n \log n + O(n)$ Swaps

[Aumüller and Dietzfelbinger(2013)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons
- $0.33n \log n + O(n)$ Swaps
- Two Pivots

[Aumüller and Dietzfelbinger(2013)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons
- $0.33n \log n + O(n)$ Swaps
- Two Pivots
  - Keeps Tracks of smalls and bigs

[Aumüller and Dietzfelbinger(2013)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons
- $0.33n \log n + O(n)$ Swaps
- Two Pivots
  - Keeps Tracks of smalls and bigs
  - Uses the information to see who to compare to first

[Aumüller and Dietzfelbinger(2013)]

Dual Pivot Quicksort with Optimal Partitioning

- $1.8n \log n + O(n)$ Comparisons
- $0.33n \log n + O(n)$ Swaps
- Two Pivots
  - Keeps Tracks of smalls and bigs
  - Uses the information to see who to compare to first
- Otherwise very similar to standard Dual Pivot Quicksort

[Aumüller and Dietzfelbinger(2013)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps
- Two Pivots

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps
- Two Pivots
  - Middle 2 of 5 elements (Evenly Spaced Out)

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps
- Two Pivots
  - Middle 2 of 5 elements (Evenly Spaced Out)
  - Uses 5-element sorting network

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps
- Two Pivots
  - Middle 2 of 5 elements (Evenly Spaced Out)
  - Uses 5-element sorting network
- Simple Simultaneous Partition algorithm

[Wild and Nebel(2012)]

Yaroslavskiy's Dual Pivot Quicksort

- $1.9n \log n - 2.46n + O(\log(n)$ Comparisons
- $0.6n \log n + 0.08n + O(\log(n))$ Swaps
- Two Pivots
  - Middle 2 of 5 elements (Evenly Spaced Out)
  - Uses 5-element sorting network
- Simple Simultaneous Partition algorithm
- Two Recursive Calls

[Wild and Nebel(2012)]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons
- $0.615n \log n + O(n)$ Swaps

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons
- $0.615n \log n + O(n)$ Swaps
- Three Pivots

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons
- $0.615n \log n + O(n)$ Swaps
- Three Pivots
  - Middle 3 of 7 elements (Evenly Spaced Out)

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons
- $0.615n \log n + O(n)$ Swaps
- Three Pivots
  - Middle 3 of 7 elements (Evenly Spaced Out)
- Simultaneous Partition algorithm

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

- $1.846n \log n + O(n)$ Comparisons
- $0.615n \log n + O(n)$ Swaps
- Three Pivots
  - Middle 3 of 7 elements (Evenly Spaced Out)
- Simultaneous Partition algorithm
- Four Recursive Calls

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons
- $O(n \log n)$ Swaps

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons
- $O(n \log n)$ Swaps
- M Pivots

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons
- $O(n \log n)$ Swaps
- M Pivots
  - Sort $2M$ elements

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons
- $O(n \log n)$ Swaps
- M Pivots
  - Sort $2M$ elements
- Partition each segment on at a time

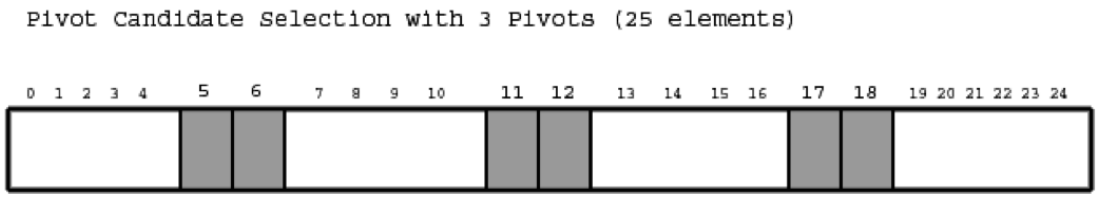
[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

M-Pivot Quicksort

- $O(n \log n)$ Comparisons
- $O(n \log n)$ Swaps
- M Pivots
  - Sort $2M$ elements
- Partition each segment on at a time
- $M + 1$ Recursive Calls

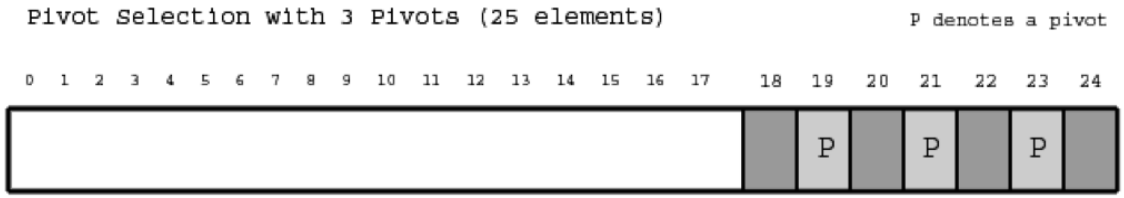
[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Example M-Pivot Selection and Partitioning



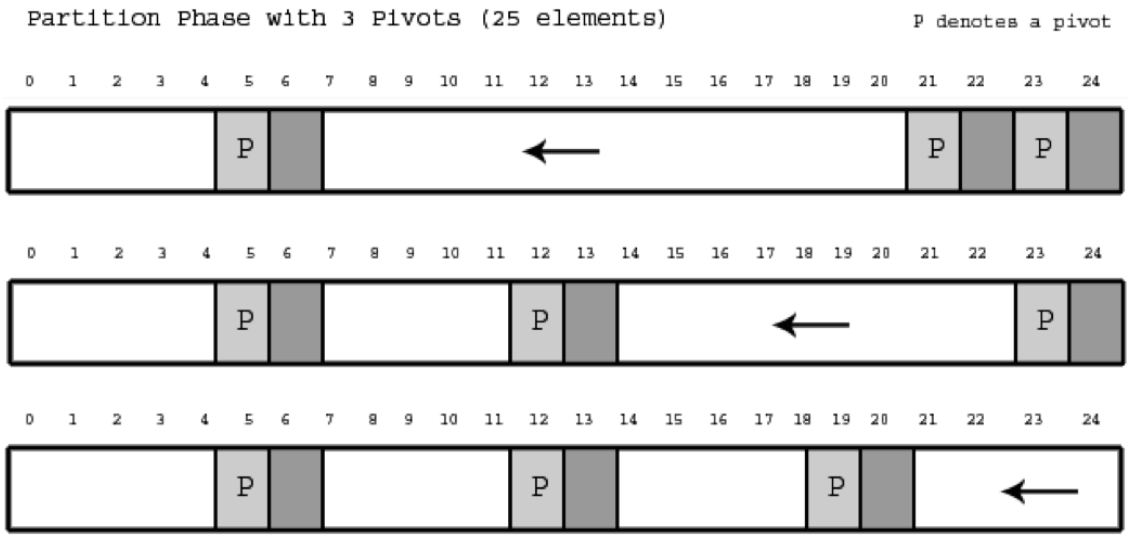
[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Example M-Pivot Selection and Partitioning



[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Example M-Pivot Selection and Partitioning



[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Heap Optimized M-Pivot Quicksort

- Enforce the Heap Property to the array before any computation

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Heap Optimized M-Pivot Quicksort

- Enforce the Heap Property to the array before any computation
  - Adds only $O(n)$ run time at each call

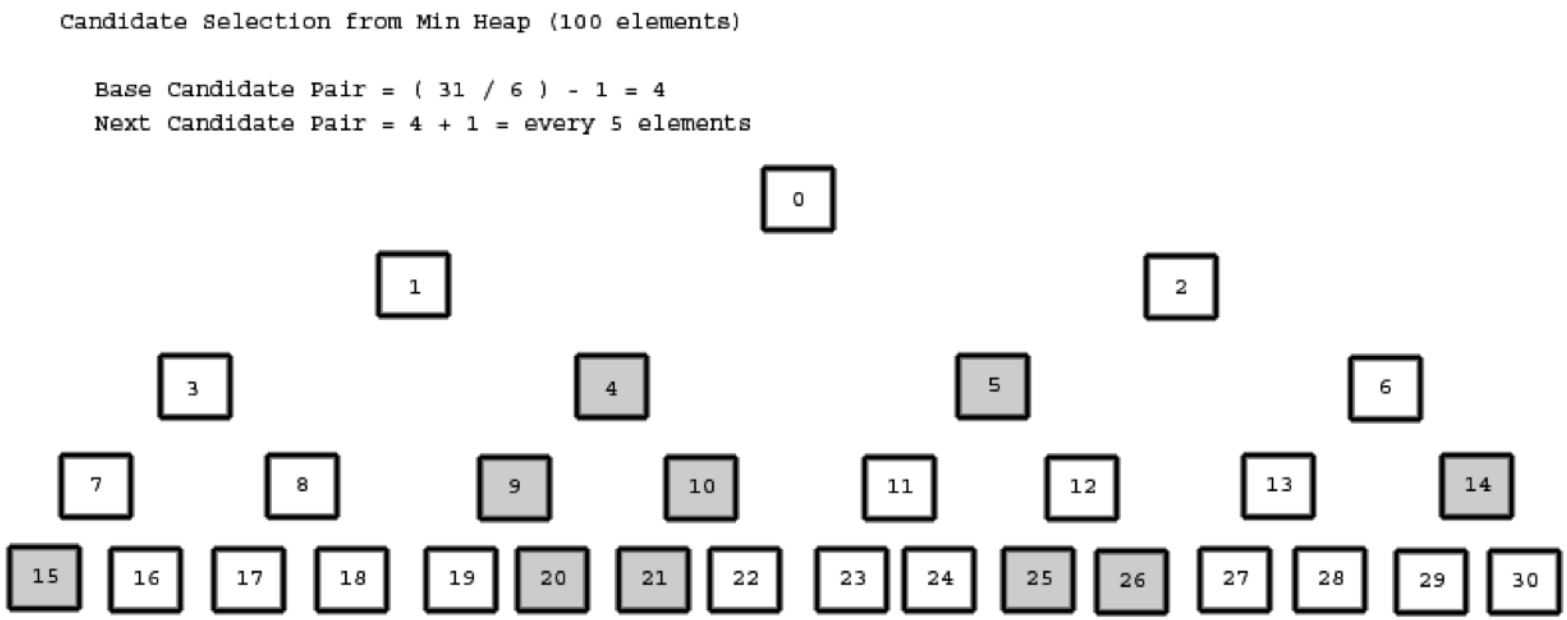
[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Heap Optimized M-Pivot Quicksort

- Enforce the Heap Property to the array before any computation
  - Adds only $O(n)$ run time at each call
  - Improves the sort to optimize pivot selection

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Example M-Pivot Heap Optimized Pivot Selection



[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Theoretical Average Case Run Time

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	$O(n \log n)$

[Aumüller and Dietzfelbinger(2013)]

[Wild and Nebel(2012)]

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

Theoretical Average Case Run Time

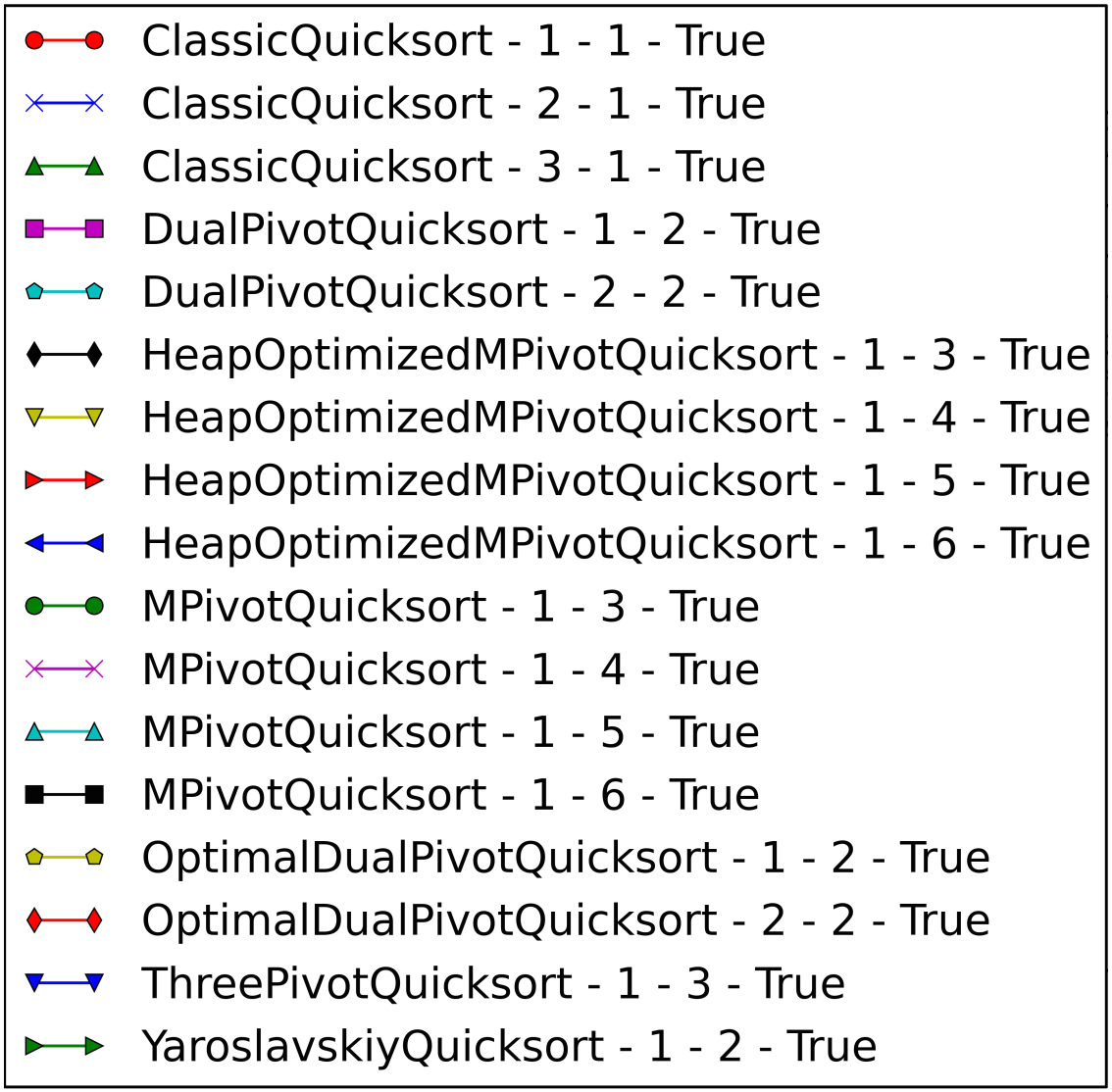
Sort Method	Swaps
Classic	$0.33n \log n - 0.58n + O(\log(n))$
Dual Pivot	$0.8n \log n - 0.3n + O(\log(n))$
Optimal Dual Pivot	$0.33n \log n + O(n)$
Three Pivot	$0.615n \log n + O(n)$
Yaroslavskiy	$0.6n \log n + 0.08n + O(\log(n))$
M Pivot	$O(n \log n)$

[Aumüller and Dietzfelbinger(2013)]

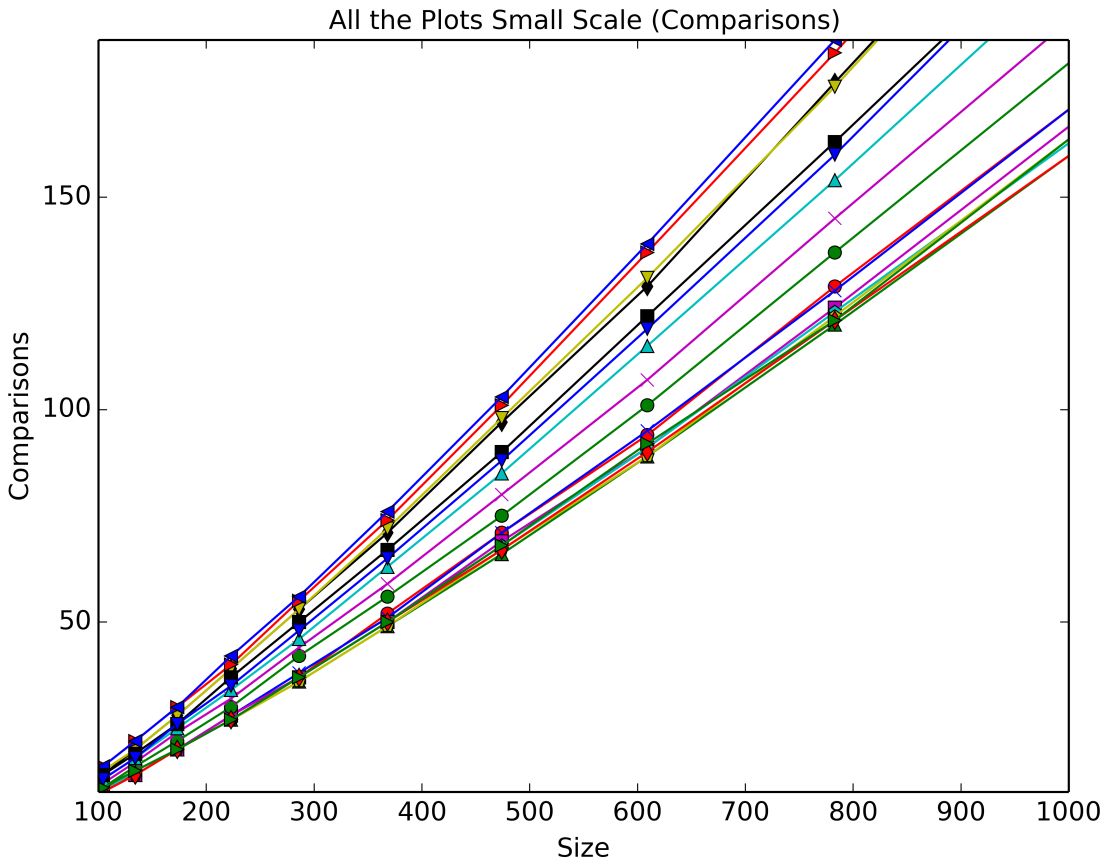
[Wild and Nebel(2012)]

[Kushagra et al.(2013)Kushagra, López-Ortiz, Qiao, and Munro]

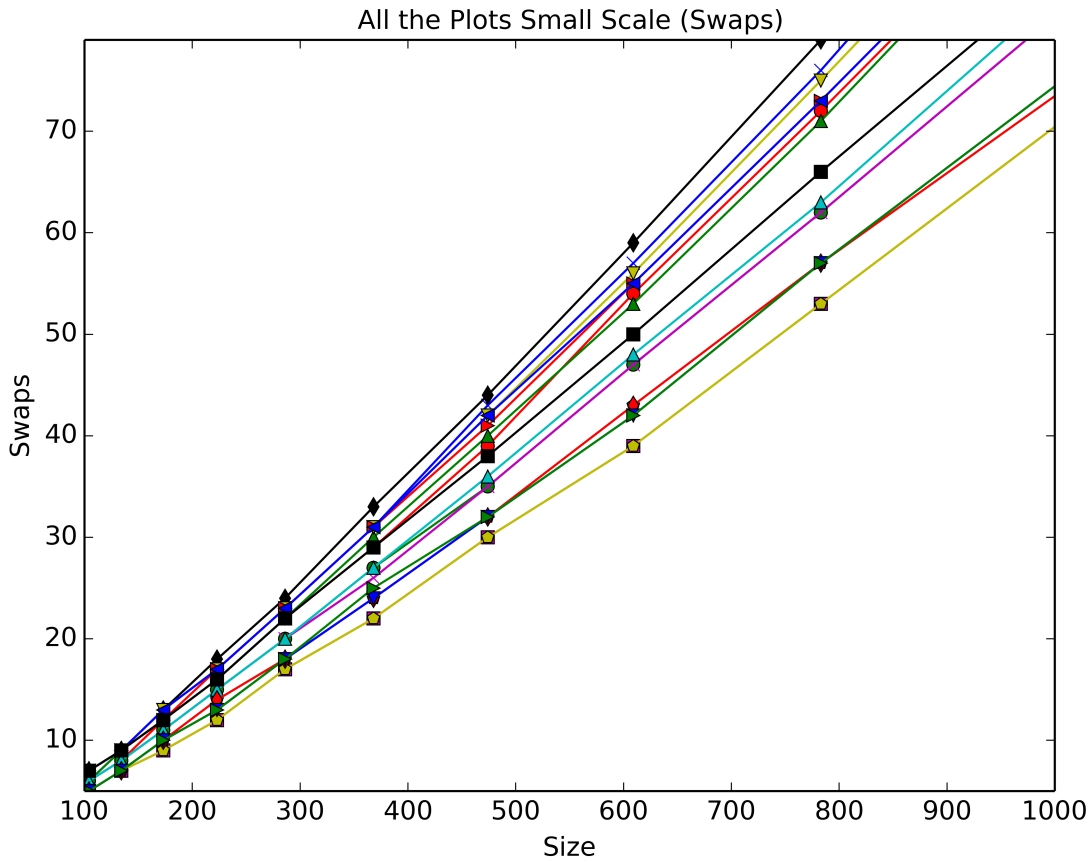
Legend



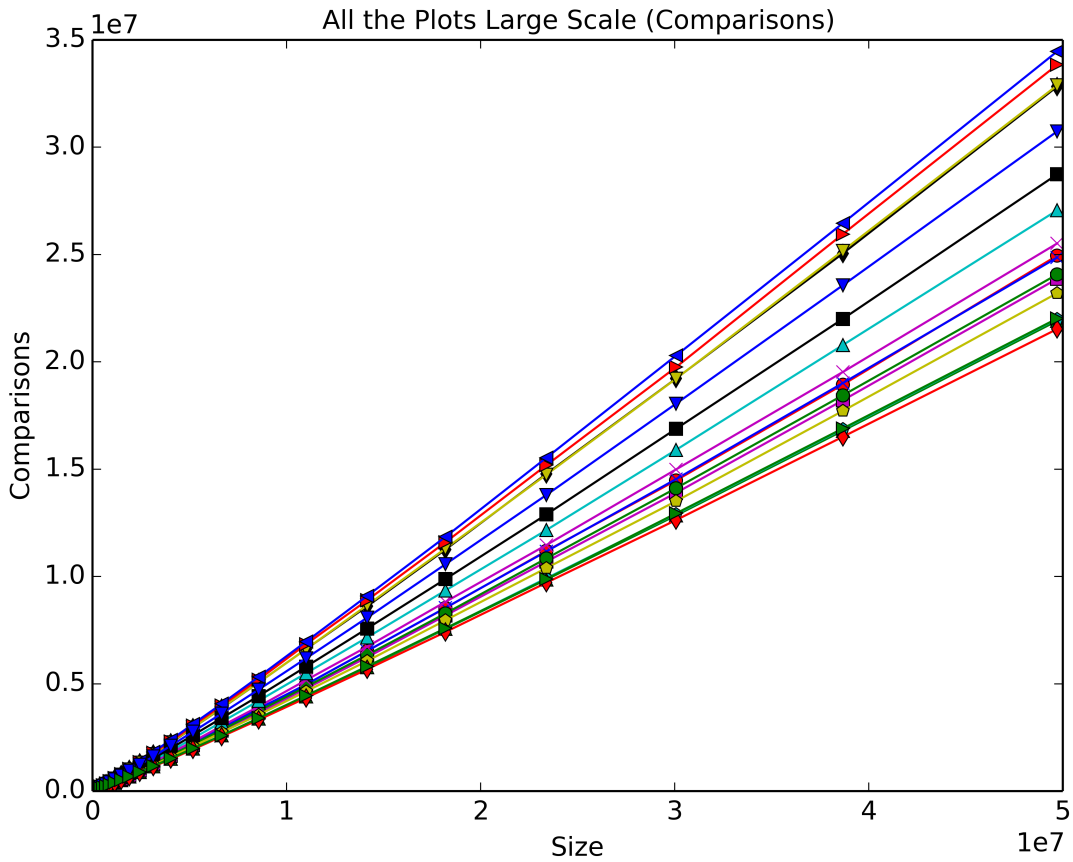
Mass Comparison Small Scale



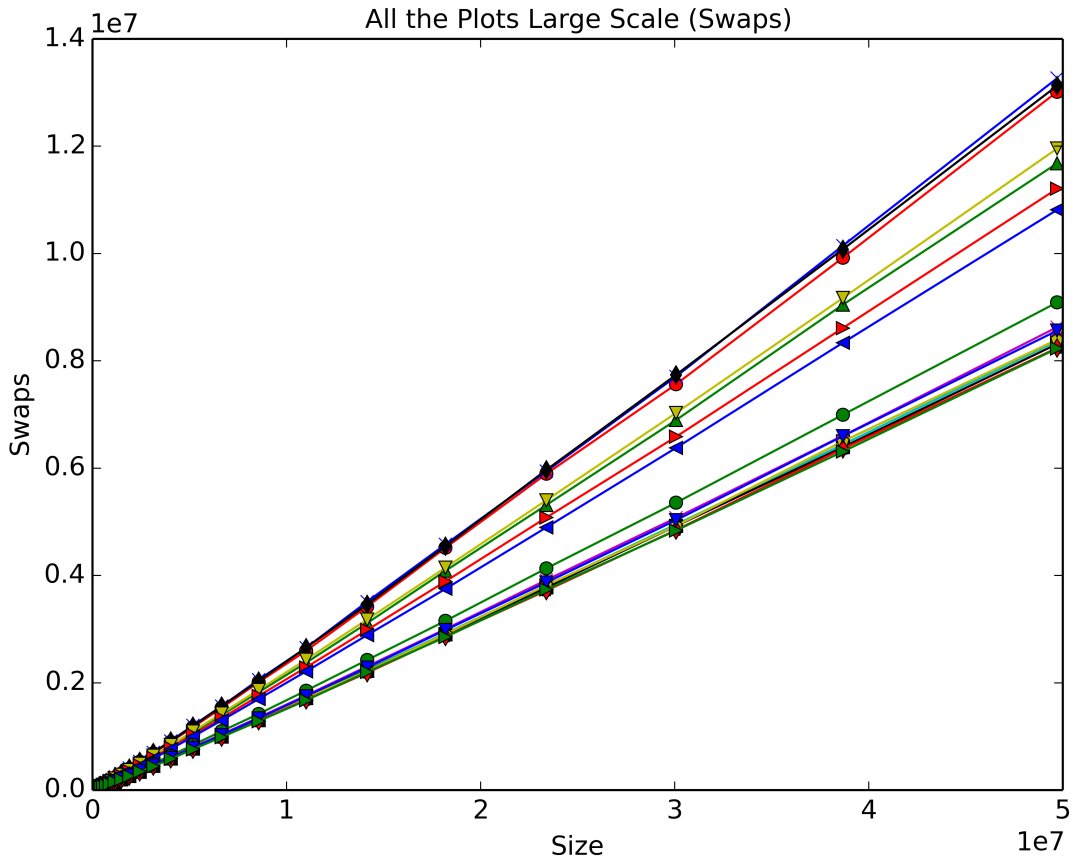
Mass Comparison Small Scale



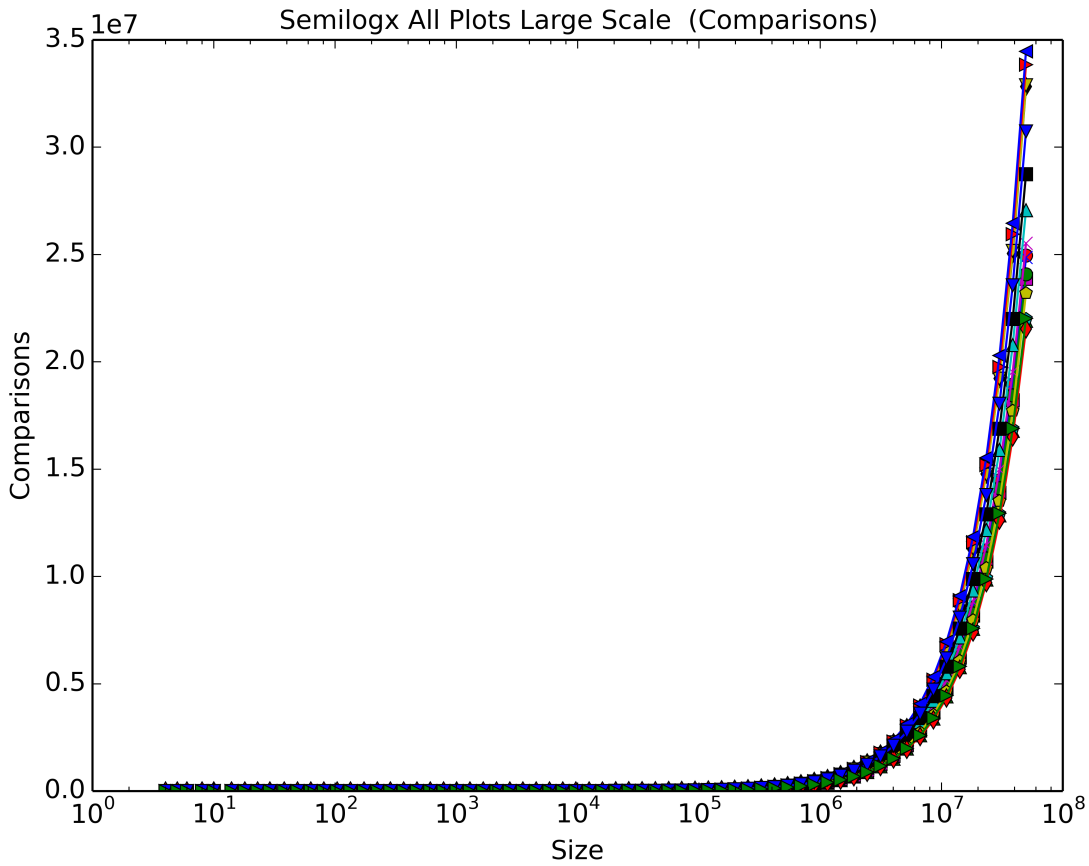
Mass Comparison Large Scale



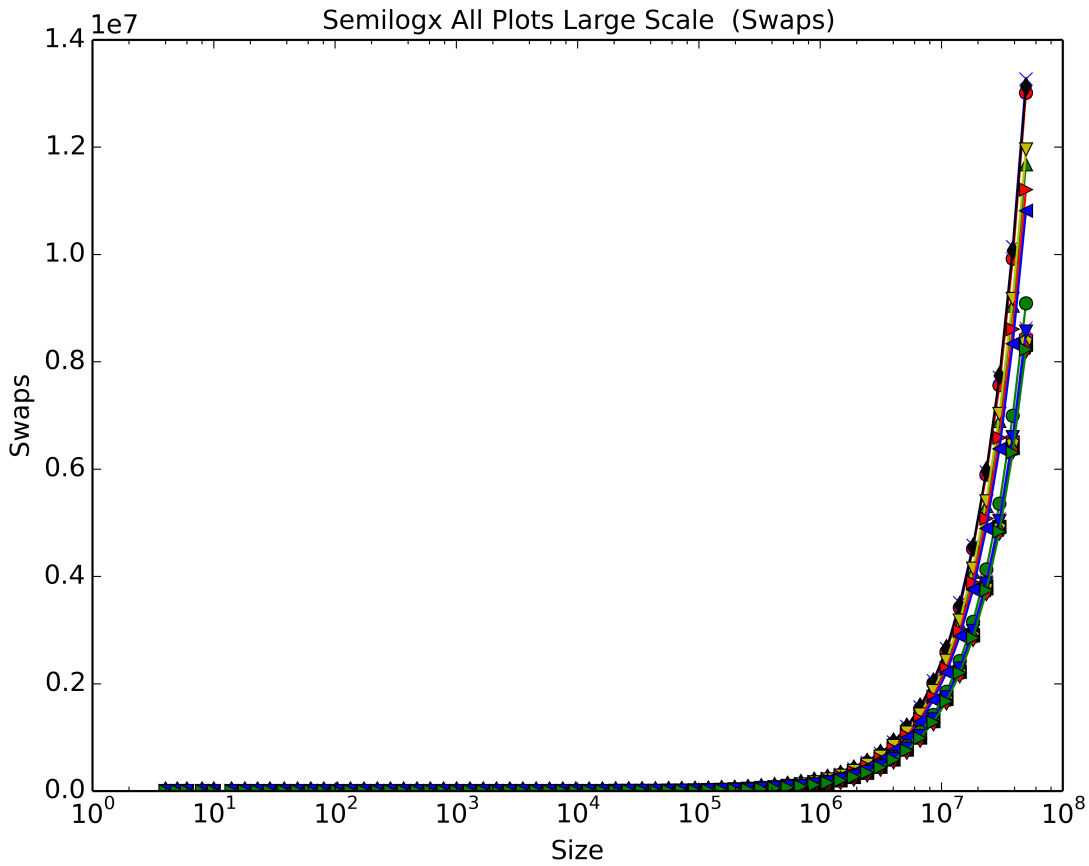
Mass Comparison Large Scale



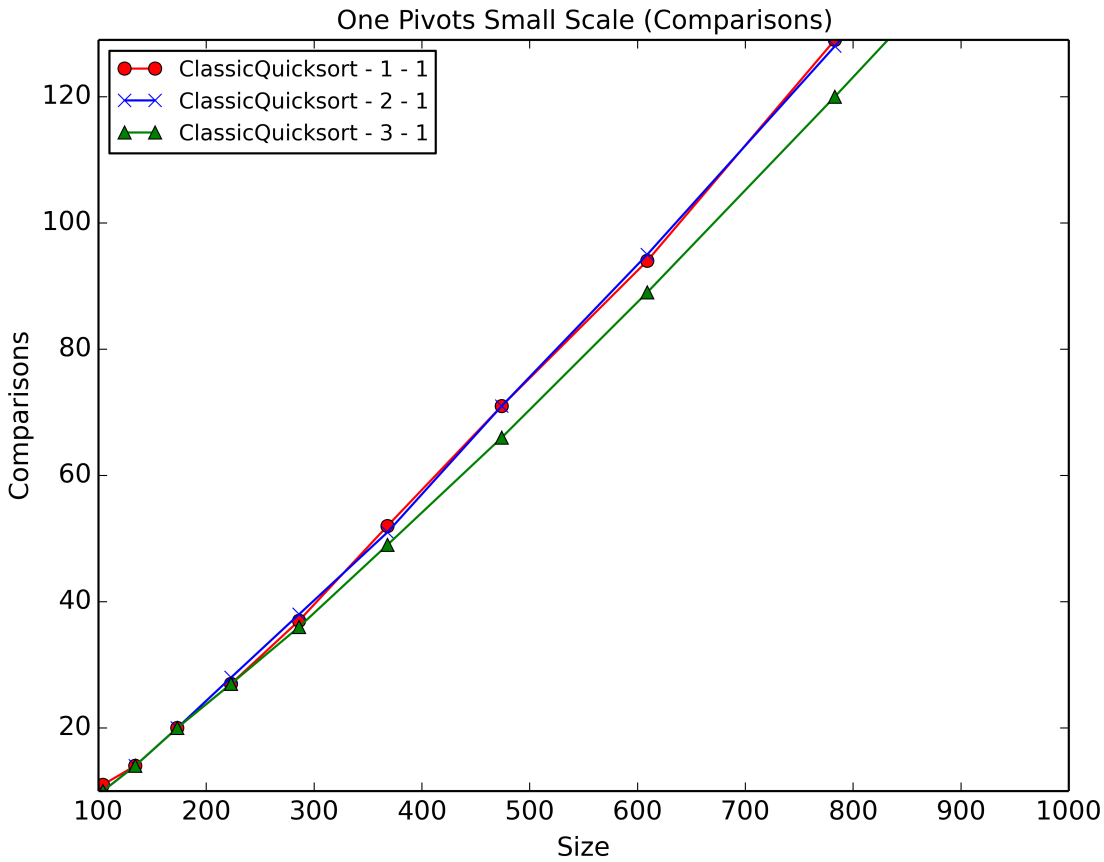
Mass Comparison Semi Log x



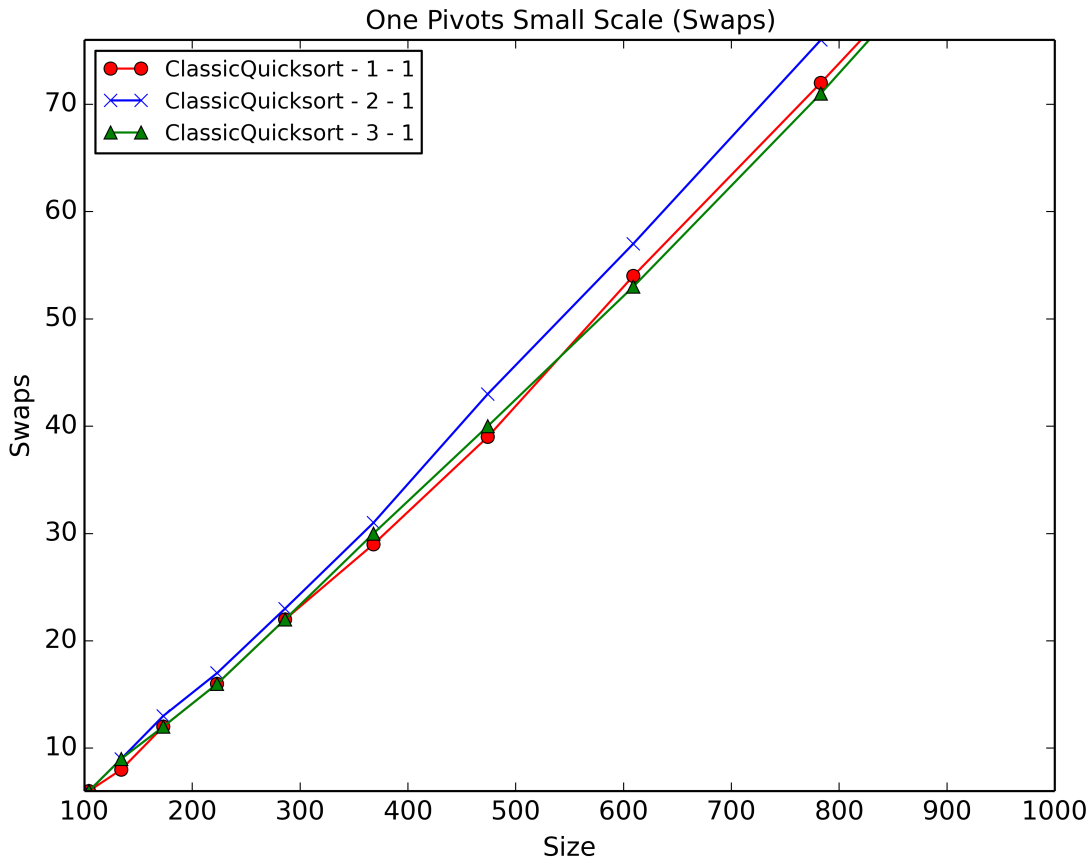
Mass Comparison Semi Log x



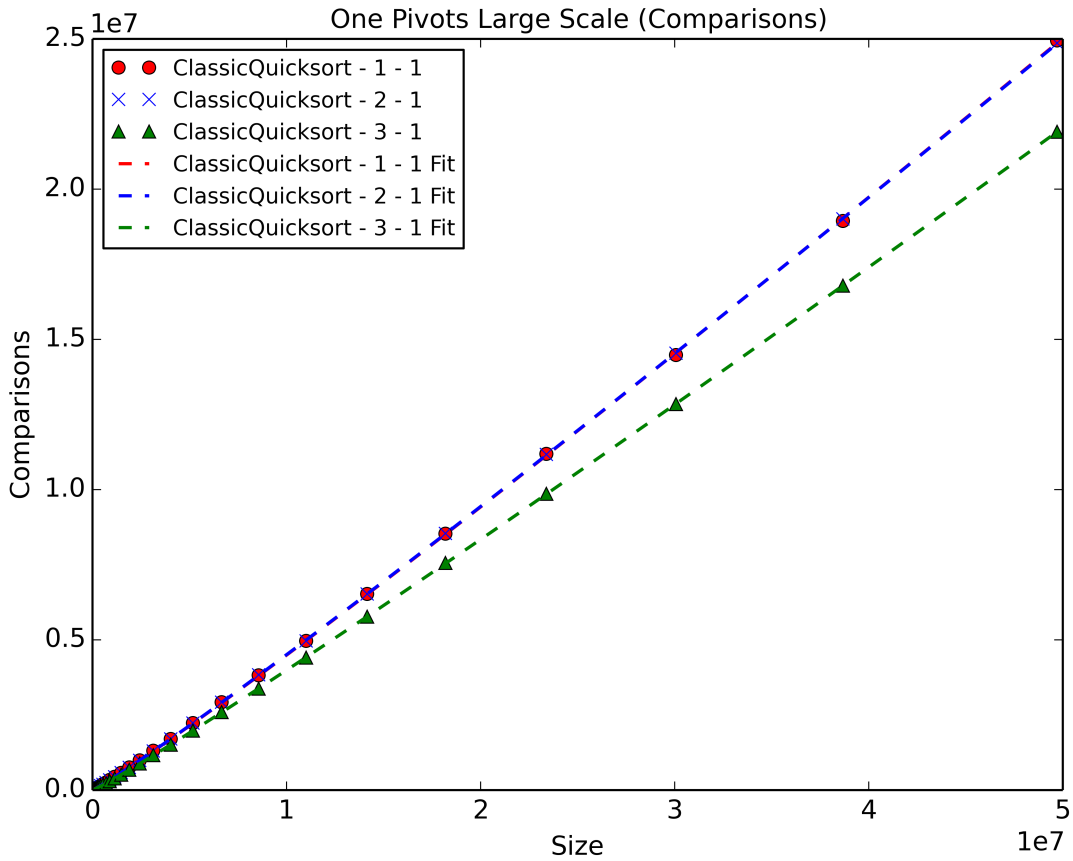
One Pivot Comparison Small Scale



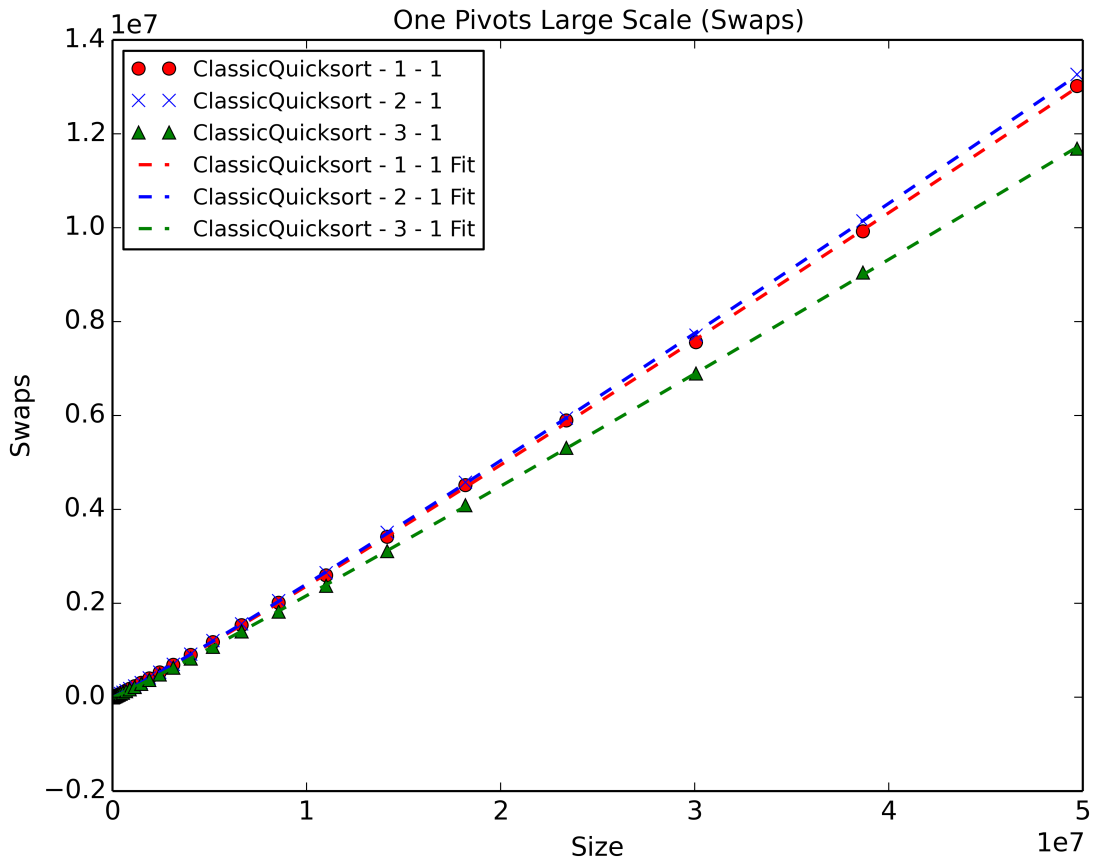
One Pivot Comparison Small Scale



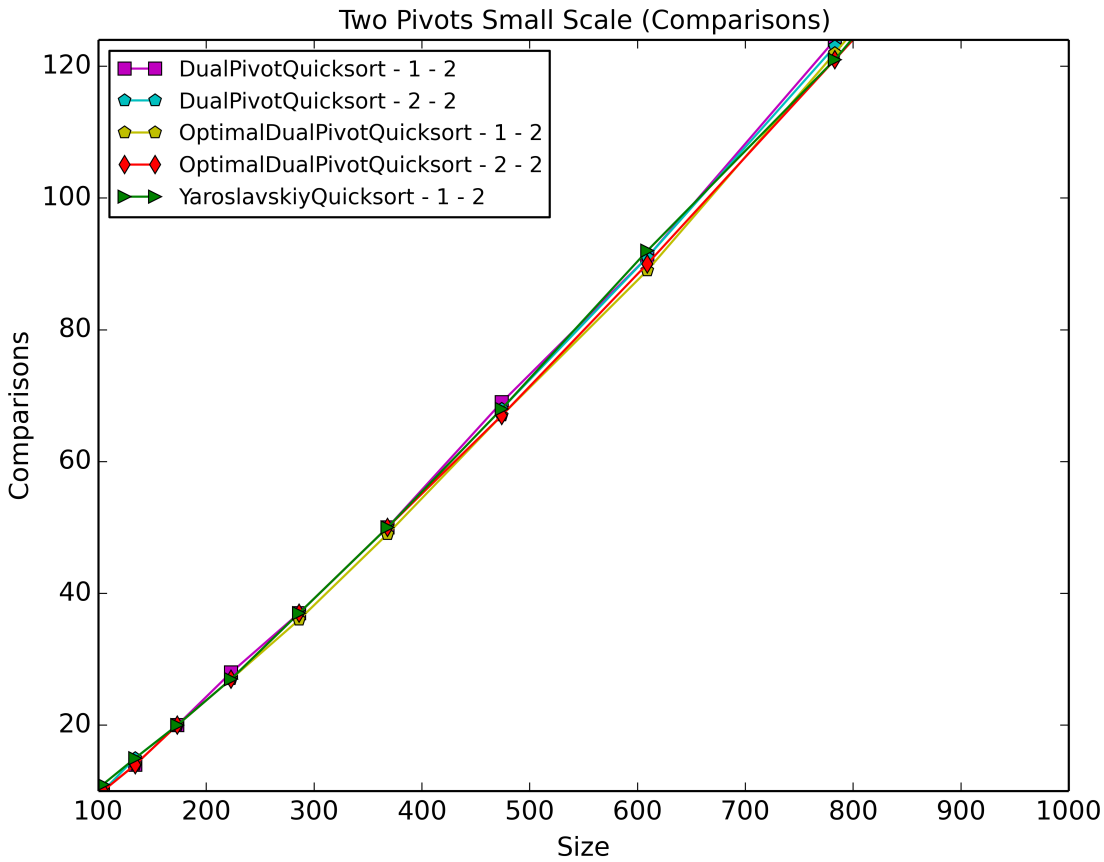
One Pivot Comparison Large Scale



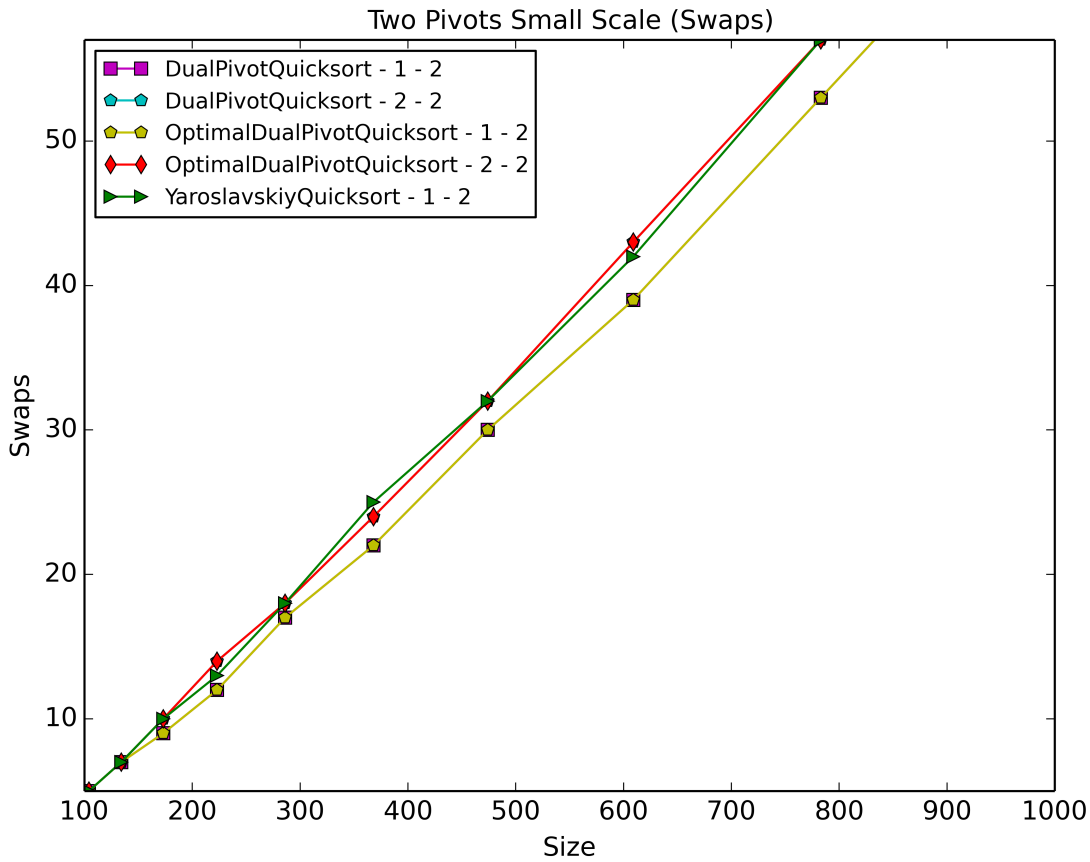
One Pivot Comparison Large Scale



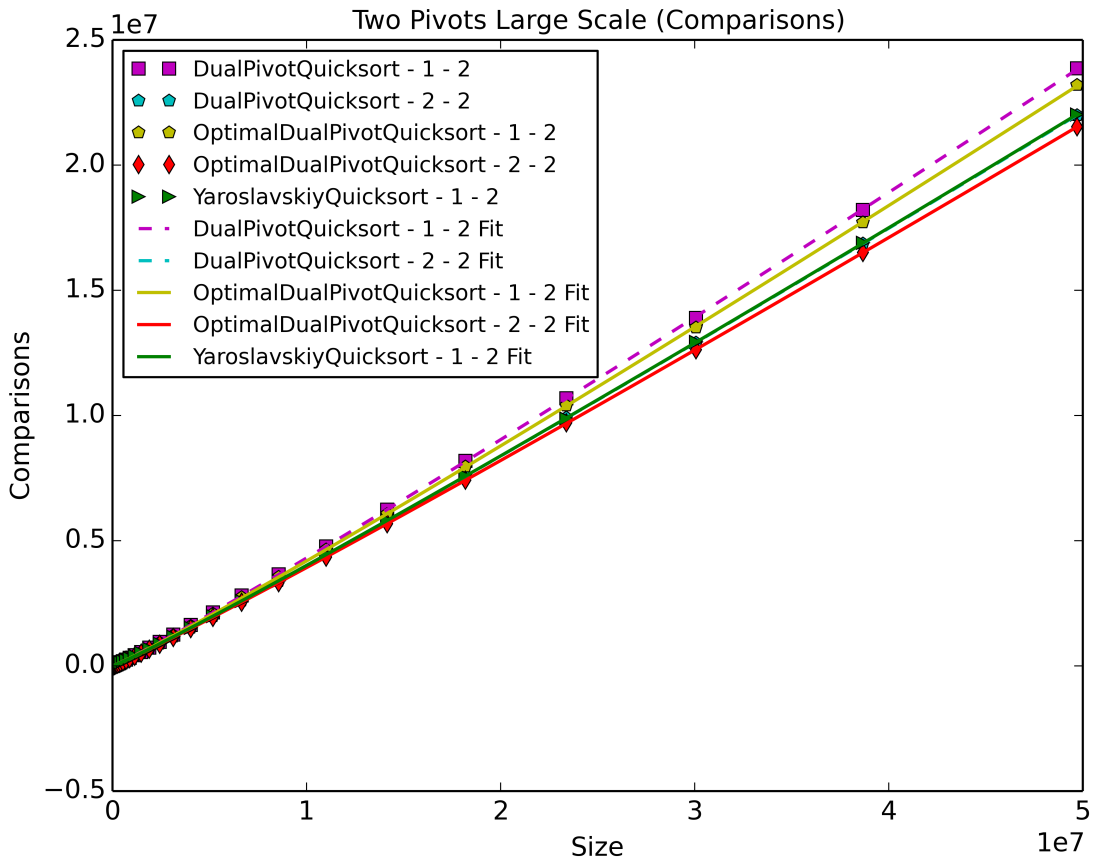
Two Pivot Comparison Small Scale



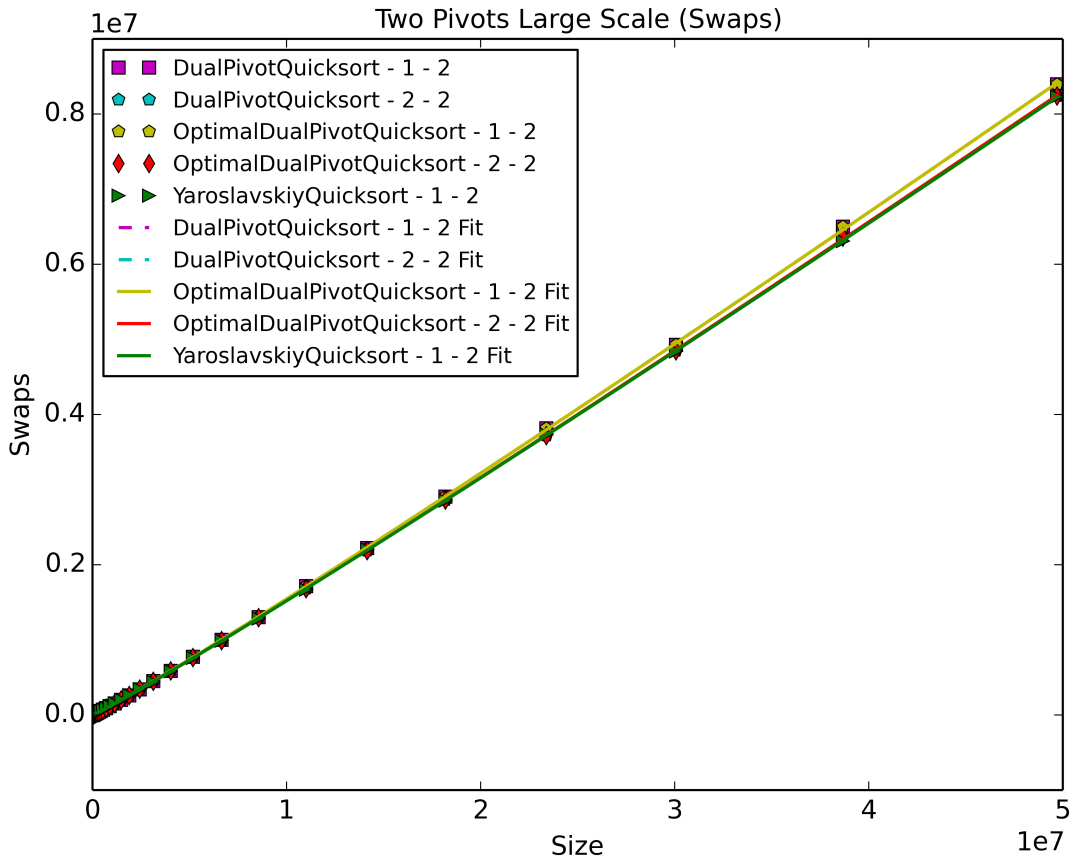
Two Pivot Comparison Small Scale



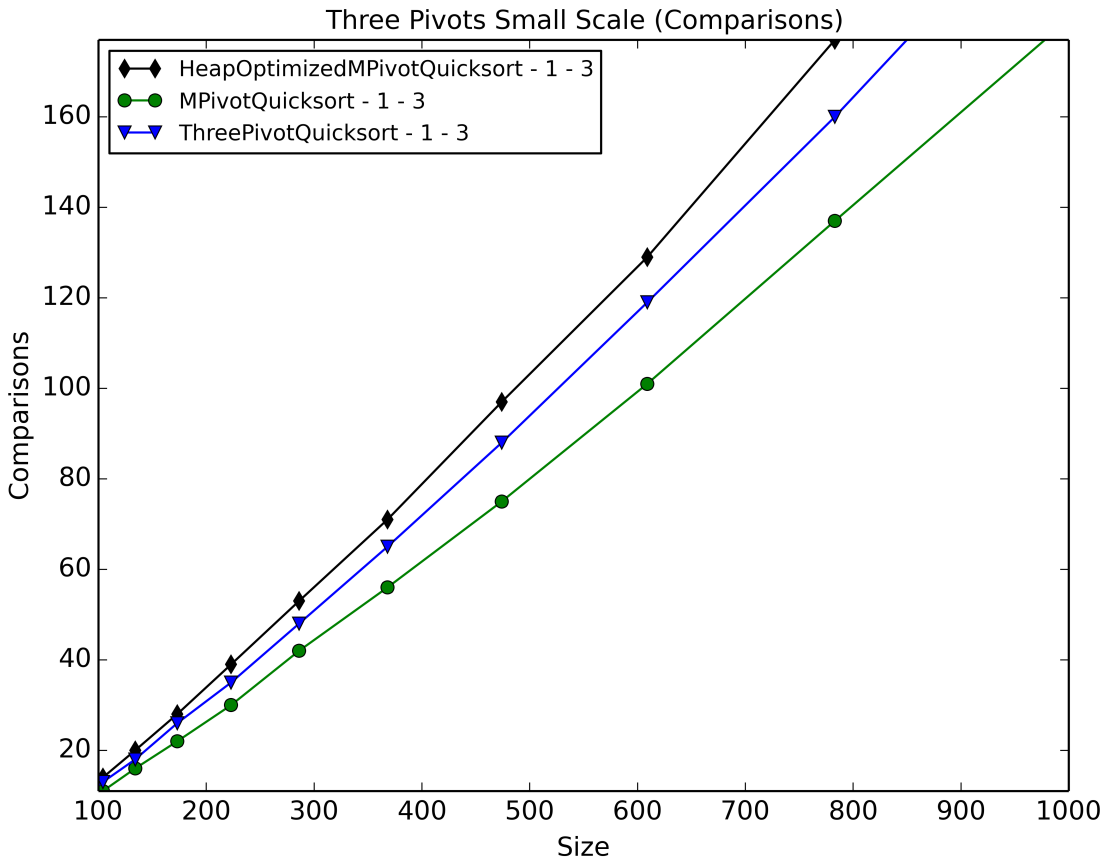
Two Pivot Comparison Large Scale



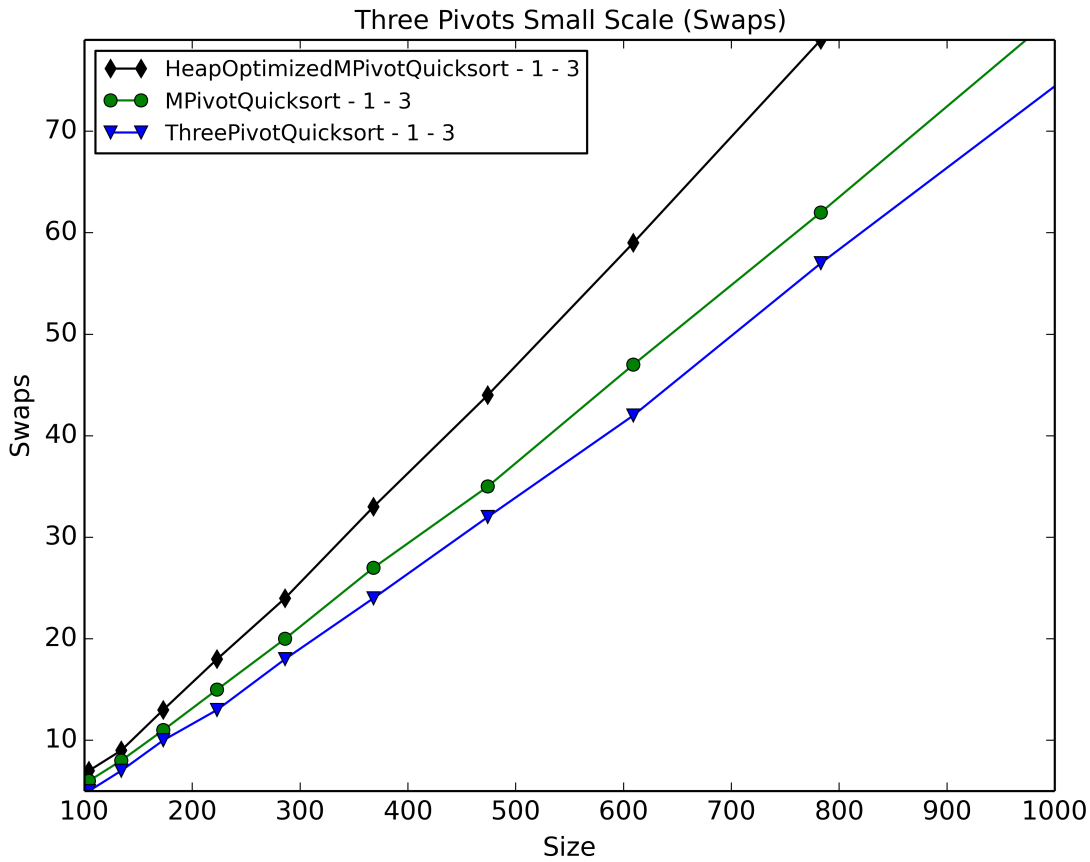
Two Pivot Comparison Large Scale



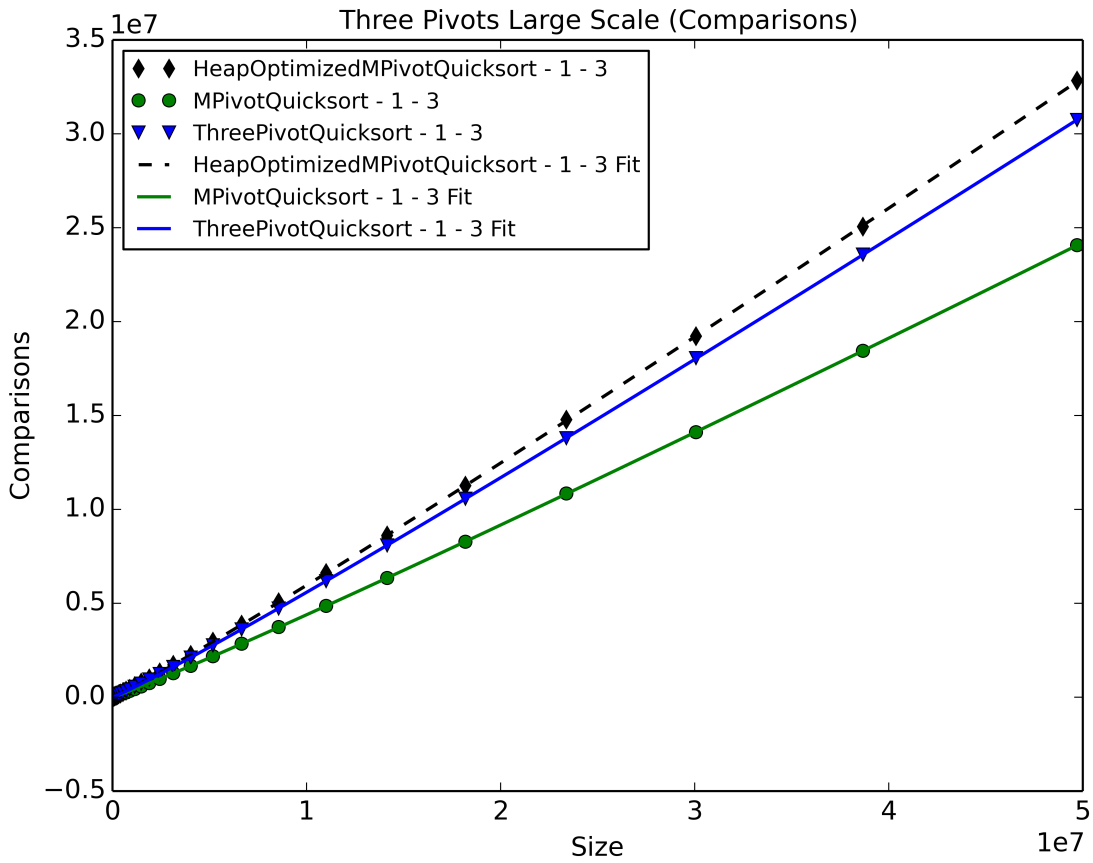
Three Pivot Comparison Small Scale



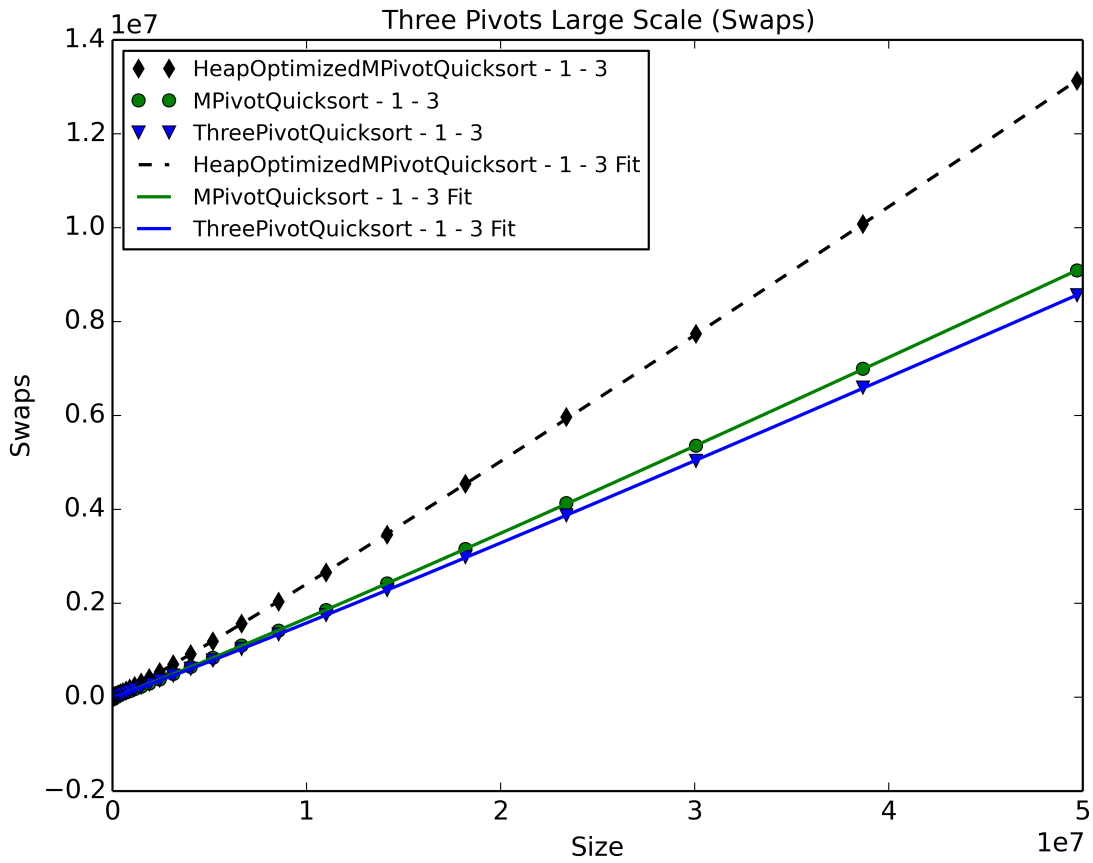
Three Pivot Comparison Small Scale



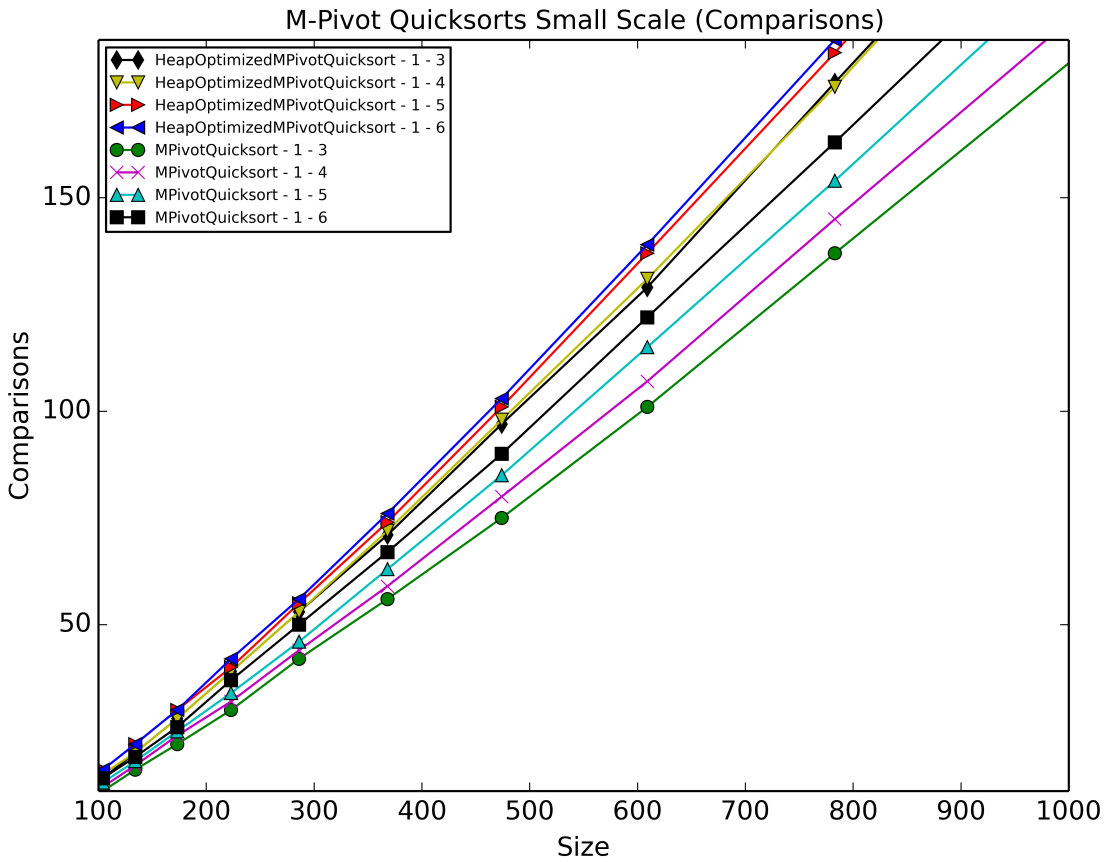
Three Pivot Comparison Large Scale



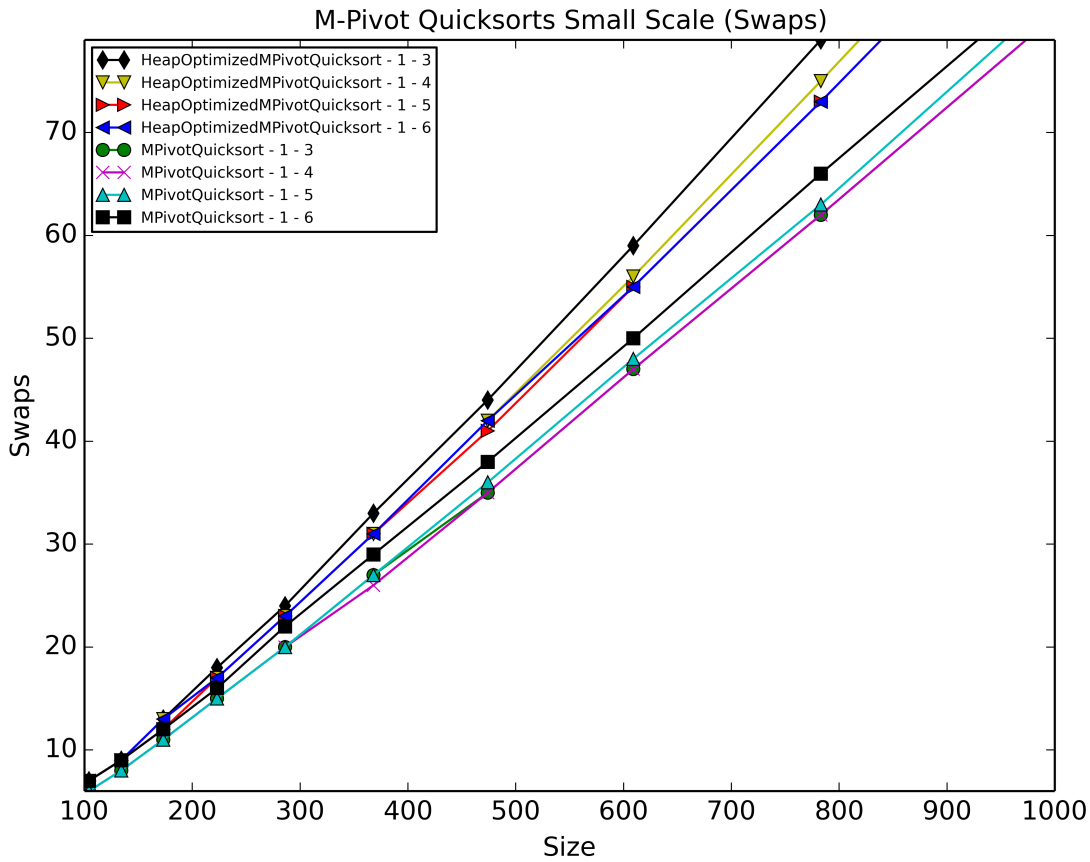
Three Pivot Comparison Large Scale



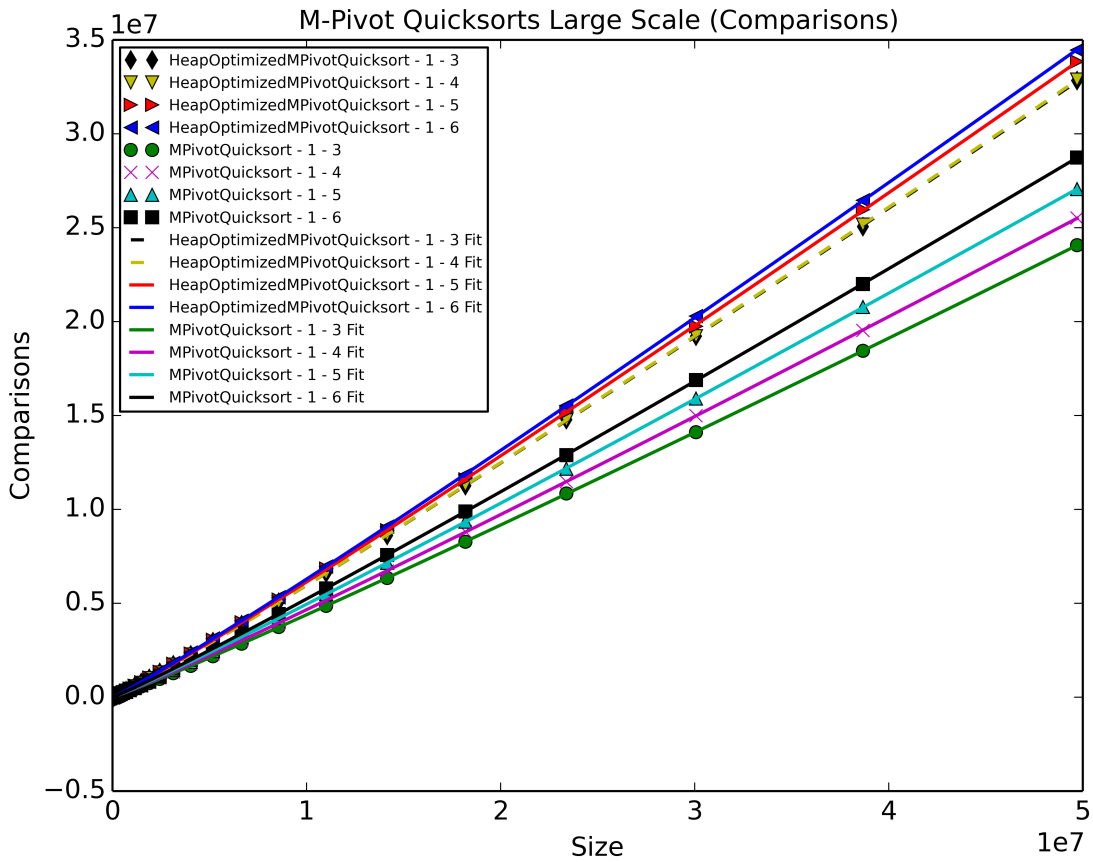
M Pivot Comparison Small Scale



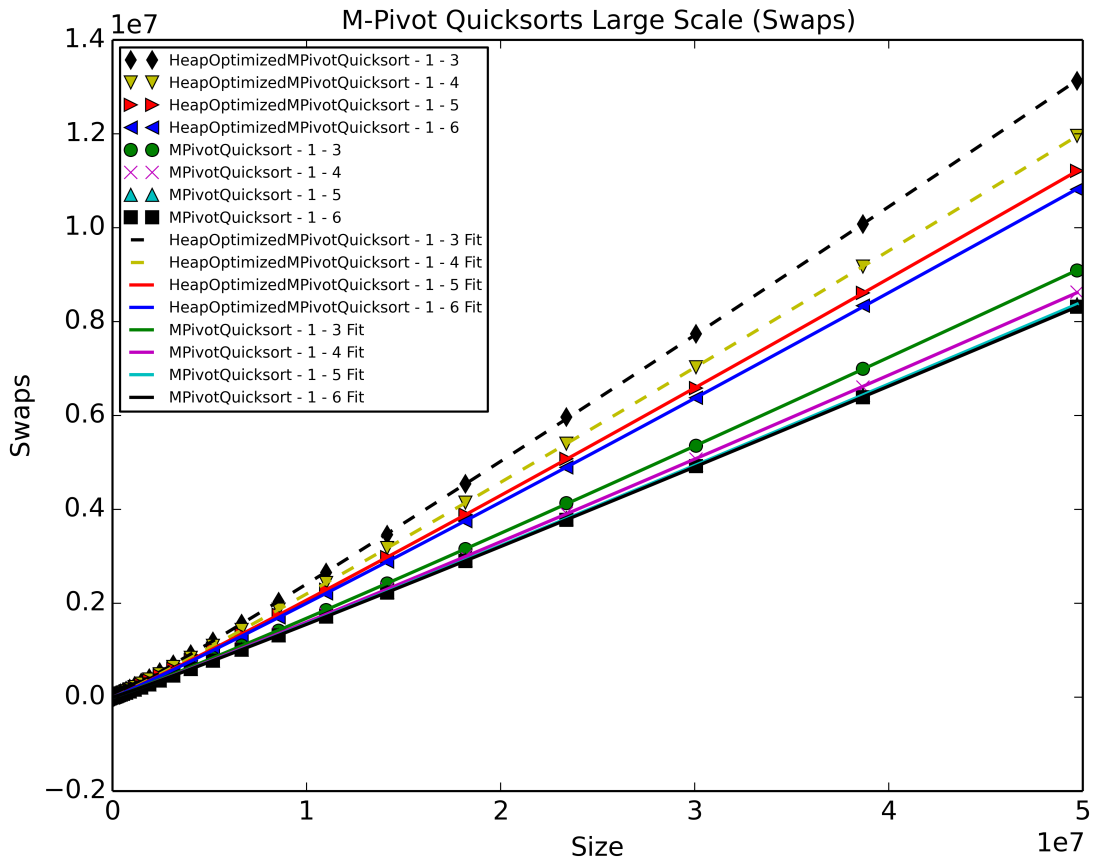
M Pivot Comparison Small Scale



M Pivot Comparison Large Scale



M Pivot Comparison Large Scale



Fit Coefficients of $An\log(n) + Bn + C\log(n)$

Sort Method	Comparisons	Swaps
Classic - 1 - 1	0.02219	0.01060
Classic - 2 - 1	0.02126	**0.01110**
Classic - 3 - 1	0.01799	0.00828
Dual Pivot - 1 - 2	0.02109	0.00636
Dual Pivot - 2 - 2	0.01787	0.00603
Optimal Dual Pivot - 1 - 2	0.02044	0.00636
Optimal Dual Pivot - 2 - 2	**0.01754**	0.00603
Three Pivot - 1 - 3	0.02595	0.00616
Yaroslavskiy - 1 - 2	0.01811	0.00584

Fit Coefficients of $An\log(n) + Bn + C\log(n)$

Sort Method	Comparisons	Swaps
Heap M Pivot - 1 - 3	0.02755	0.00999
Heap M Pivot - 1 - 4	0.02782	0.00885
Heap M Pivot - 1 - 5	**0.02903**	0.00809
Heap M Pivot - 1 - 6	0.02801	0.00769
M Pivot - 1 - 3	0.01955	0.00640
M Pivot - 1 - 4	0.02039	0.00594
M Pivot - 1 - 5	0.02136	0.00532
M Pivot - 1 - 6	0.02369	**0.00524**

References I

Martin Aumüller and Martin Dietzfelbinger.
Optimal partitioning for dual pivot quicksort.
In *Proceedings of the 40th International Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'13, pages 33–44, Berlin, Heidelberg, 2013. Springer-Verlag.
ISBN 978-3-642-39205-4.
doi: 10.1007/978-3-642-39206-1_4.
URL
http://dx.doi.org/10.1007/978-3-642-39206-1_4.

Shrinu Kushagra, Alejandro López-Ortiz, Aurick Qiao, and J Ian Munro.
Multi-pivot quicksort: Theory and experiments.
2013.

References II

Sebastian Wild and Markus E. Nebel.
Average case analysis of java 7's dual pivot quicksort.
In *Proceedings of the 20th Annual European Conference on Algorithms*, ESA'12, pages 825–836, Berlin, Heidelberg, 2012. Springer-Verlag.
ISBN 978-3-642-33089-6.
doi: 10.1007/978-3-642-33090-2_71.
URL
http://dx.doi.org/10.1007/978-3-642-33090-2_71.

Questions?