Introduction

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

Data is partitioned By Comparing and Swapping elements

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

Data is partitioned By Comparing and Swapping elements

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

Data is partitioned By Comparing and Swapping elements

Introduction

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

Data is partitioned By Comparing and Swapping elements

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

Data is partitioned By Comparing and Swapping elements

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

Data is partitioned By Comparing and Swapping elements

Dual Pivot Quicksort

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

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

- $2n\log n - 1.51n + O(\log(n))$ Comparisons
- $0.8n\log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots

[?]

Tertile Elements $=$ Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

- $2n \log n - 1.51n + O(\log(n))$ Comparisons
- $0.8n \log n - 0.3n + O(\log(n))$ Swaps
- Two Pivots
  - First and Last Element

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

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

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

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

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Dual Pivot Quicksort

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

[?]

Tertile Elements = Middle 2 of 5 elements (Evenly Spaced Out)

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

Kushagra-Ortiz-Qiao-Munro Tri-Pivot Quicksort

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

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

[?]

This algorithm had incomplete pseudo-code. Only the partition algorithm was specified.

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	$O(n \log n)$

[?]

[?]

[?]

These are average case run times. The M-Pivot quicksort is not studied very well. As you see, people study each quicksort with a fixed number of pivots.

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	$O(n \log n)$

[?]

[?]

[?]

These are average case run times. The M-Pivot quicksort is not studied very well. As you see, people study each quicksort with a fixed number of pivots.

Will add slides to answer potential questions