Sorting

David Croft

Introduction

Stable sort

Selection sort

algorithms

QUICKSOFT

Divide & Conquer

Divide & Conquer

Recap

Sorting algorithms

David Croft

Coventry University david.croft@coventry.ac.uk

March 1, 2016



Bubblesort
Stable sort
In-place

Selection sort

algorithms

Divide & Conqu

Comparing

Docan

1 Introduction

- 2 Bubblesort
 - Stable sort
 - In-place
- 3 Selection sort
- 4 Other algorithms
- 5 Quicksort
 - Divide & Conquer
- 6 Comparing
- 7 Recap



Bubblesort
Stable sort
In-place

Selection sort

Other algorithm:

Quicksort

Divide & Conqui

Companing

Sorting is one of the classic problems for learning algorithms.

- Requirement for everything.
- Obvious applications like sorting text, statistics (median calculations).
- Less obvious, sorting objects in games for FOV (Field Of View) calculations.
- Route planning.



niti oduction

Stable sort
In-place

Selection sort

Other algorithms

Divide & Conqu

Comparing

Recap

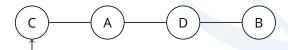
- Compares each item to the next in the sequence.
 - Swap items if in wrong order.



Very simple sort.

Compares each item to the next in the sequence.

Swap items if in wrong order.



Stable Soft

Selection sol

Other algorithms

Divide & Conque

Comparing

Recap





Bubblesort Stable sort

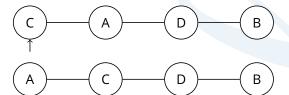
Selection sort

Other algorithm:

Quicksort

Recap

- Compares each item to the next in the sequence.
 - Swap items if in wrong order.





Bubblesort
Stable sort

Selection sort

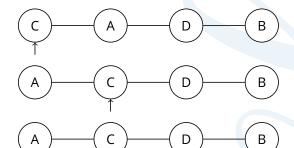
Other algorithms

Divide & Conque

Compari

Recap

- Compares each item to the next in the sequence.
 - Swap items if in wrong order.







Bubblesort Stable sort

Selection sort

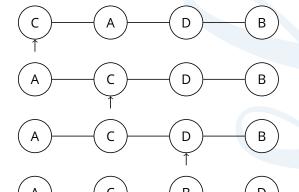
Other algorithms

Divide & Conque

Commovin

Recap

- Compares each item to the next in the sequence.
 - Swap items if in wrong order.





Introduction

Bubblesort Stable sort

Selection sort

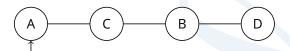
Other algorithms

algorithms

Companii

Iterating over the sequence once isn't typically enough.

■ Keep iterating over the sequence until elements are sorted.

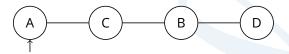




Bubblesort

Iterating over the sequence once isn't typically enough.

■ Keep iterating over the sequence until elements are sorted.



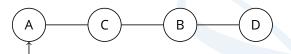


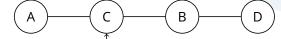


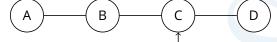
Bubblesort

Iterating over the sequence once isn't typically enough.

Keep iterating over the sequence until elements are sorted.









Bubblesort Stable sort

Selection sort

Other algorithm:

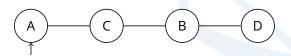
Divide & Conque

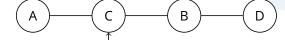
Comparing

Recap

Iterating over the sequence once isn't typically enough.

■ Keep iterating over the sequence until elements are sorted.











Bubblesor Stable sort

Selection sor

Other algorithms

Divide & Conque

Comparing

Recap

Bubble sort is what's known as a stable in-place sort.



Selection sor

Other algorithms

- Livide & Conque

Comparing

Reca

Bubble sort is what's known as a stable in-place sort.

Stable meaning that equivalent elements do not change their relative orders.

■ Not important if e.g. sorting people by height.



Selection so

algorithms

Comparing

Recan

Bubble sort is what's known as a stable in-place sort.

- Not important if e.g. sorting people by height.
- Important if e.g. priority queues.
 - Imagine a queue in an emergency room.



Bubble sort is what's known as a stable in-place sort.

- Not important if e.g. sorting people by height.
- Important if e.g. priority queues.
 - Imagine a queue in an emergency room.
 - Treat the most serious conditions first, sort people on how bad injury is.



Bubble sort is what's known as a stable in-place sort.

- Not important if e.g. sorting people by height.
- Important if e.g. priority queues.
 - Imagine a queue in an emergency room.
 - Treat the most serious conditions first, sort people on how bad injury is.
 - If many people have same injury then should be seen based on when entered queue.



Introduction
Bubblesori
Stable sort

Other

Quicksort
Divide & Conqu

Comparing Recap Bubble sort is what's known as a stable in-place sort.

Stable meaning that equivalent elements do not change their relative orders.

- Not important if e.g. sorting people by height.
- Important if e.g. priority queues.
 - Imagine a queue in an emergency room.
 - Treat the most serious conditions first, sort people on how bad injury is.
 - If many people have same injury then should be seen based on when entered queue.

With unstable sorting algorithm the relative orders of equivalent elements can be changed.



Selection sor

algorithms

Comparing

Recap

In-place meaning that it only needs a small amount of additional memory in order to work.

- More memory efficient than the alternative.
- Can be important if...
 - ...dealing with large amounts of data.
 - ...have limited resources (i.e. embedded systems).
- Bubble sort only needs a few extra variables to swap the elements and to step through the sequence.



Bubblesor Stable sort

Selection sor

Other algorithm:

Divide & Conque

Comparing

Recan

- Explained here to introduce you to sorting concepts.
 - In-place, stable.



Bubblesor Stable sort

Selection sor

Other algorithm:

Divide & Conque

Camanarina

Recap

- Explained here to introduce you to sorting concepts.
 - In-place, stable.
- Is rubbish.



Bubblesor Stable sort In-place

Selection sort

Other algorithms

Divide & Conque

Comparir

Recap

- Explained here to introduce you to sorting concepts.
 - In-place, stable.
- Is rubbish.
 - Horrible performance, average is $O(n^2)$.



Bubblesort
Stable sort
In-place

Selection sor

Other algorithms

Divide & Conquer

Comparir

Recan

- Explained here to introduce you to sorting concepts.
 - In-place, stable.
- Is rubbish.
 - Horrible performance, average is $O(n^2)$.
 - But best case is only O(n).



Selection so

algorithms

Divide & Conque

Comparing

Recap

The time taken to sort a sequence depends on:

■ The starting order of the sequence.

For example, Bubblesorting a 100 elements:



Other

Other algorithms

Comparing

The time taken to sort a sequence depends on:

■ The starting order of the sequence.

For example, Bubblesorting a 100 elements:

- Best case, are already sorted.
 - Iterate over sequence once.
 - 100 comparisons.



Other

algorithms

Comparing

Recap

The time taken to sort a sequence depends on:

■ The starting order of the sequence.

For example, Bubblesorting a 100 elements:

- Best case, are already sorted.
 - Iterate over sequence once.
 - 100 comparisons.
- Worst case, in reverse order.
 - Iterate over sequence 100 times.
 - 10,000 comparisons.



Other

algorithms

Comparing

Reca

The time taken to sort a sequence depends on:

■ The starting order of the sequence.

For example, Bubblesorting a 100 elements:

- Best case, are already sorted.
 - Iterate over sequence once.
 - 100 comparisons.
- Worst case, in reverse order.
 - Iterate over sequence 100 times.
 - 10,000 comparisons.
- Average case, random order.
 - Somewhere in between.





Selection sort

- Divides sequence into sorted and unsorted regions.
- Not stable.
- In place.
- Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.



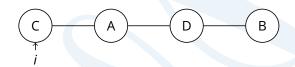
Bubblesort Stable sort In-place

Selection sort

algorithm

Quicksort
Divide & Congu

Comparing



- Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.



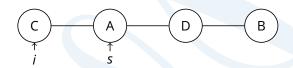
Bubblesor Stable sort In-place

Selection sort

algorithm

Divide & Conque

Comparing



- 1 Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.



Bubblesor Stable sort

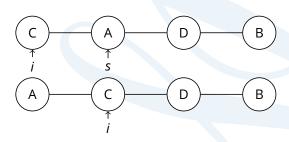
Selection sort

algorithm

Divide & Conqu

Comparing

- Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.





Bubblesor Stable sort

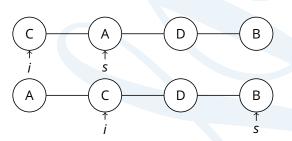
Selection sort

algorithm

Divide & Conqu

Comparing

- Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.





Bubblesor Stable sort

Selection sort

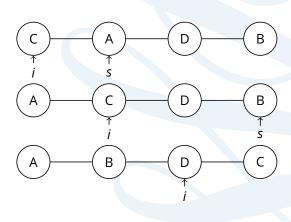
algorith

Quicksor

Comparing

Recai

- 1 Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.





Stable sort

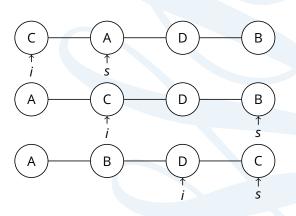
Selection sort

algorithi

Quicksor

Comparing

- 1 Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.





Bubblesort
Stable sort

Selection sort

algorith

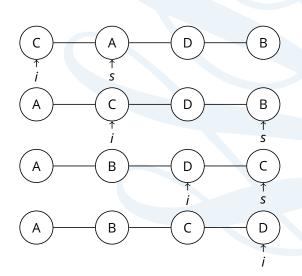
Quicksor

Divide & Cor

Comparing

Pocar

- 1 Iterate over sequence.
- For each element search the remaining elements on its right for the smallest value.
- 3 Swap smallest element with current element.





Selection sort

Bubblesort is $O(n^2)$ worst and average case. Selection sort is $O(n^2)$ worst and average case.

- Selection sort is generally faster than bubble.
 - But have same *O*() complexity.
 - What?



Introduction
Bubblesort
Stable sort
In-place

Selection sort

algorithms

Divide & Conqu

Comparin;

Bubblesort is $O(n^2)$ worst and average case . Selection sort is $O(n^2)$ worst and average case.

- Selection sort is generally faster than bubble.
 - But have same *O*() complexity.
 - What?
- \circ O() notation describes how an algorithm will grow.
- Not good at absolute performances.
- Selection sort typically does fewer comparisons and swaps than bubblesort.
 - Therefore typically faster.
- Best case bubblesort is O(n), selection is $O(n^2)$.
 - Occasionally faster.





Selection sort

Other algorithms

Quicksort
Divide & Conqu

Comparing

кеса

Many sorting algorithms

- Different trade-offs, performances. https://www.youtube.com/watch?v=ZZuD6iUe3Pc
- Some are just jokes.

- Bogo
 - Bubble
- 4 Circle
- 5 Cocktail
- 6 Comb
- Counting
- 8 Cycle

g Gnome

- 10 Неар
- 11 Insert
- 12 Merge
- 13 Pancake
- 14 Patience
- 15 Permutation
- 16 Quick

17 Radix

- 18 Selection
- 19 Shell
- 20 Sleep
- 21 Stooge
- 22 Strand
- 23 Tree





Introduction

Bubblesort Stable sort In-place

Selection sol

algorith

Quicksort
Divide & Conque

·

Comparir

Recap

Neither bubble or selection sort are very good.

- Simple algorithms but slow.
- Not used in real life.

One of the fastest sorting algorithms.

- Used in real life.
- Recursively breaks the sequence in half.
 - Divide & Conquer.



Introduction

Bubbleson Stable sort

Selection sor

algorithm

Quicksort
Divide & Conque

Divide & Conque

Comparing

Recap

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.



Bubblesort Stable sort In-place

Selection sort

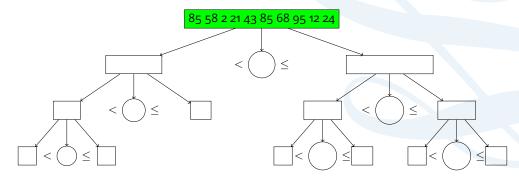
algorithm

Quicksort
Divide & Conque

. Comparing

Recap

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Bubblesort
Stable sort

Selection sort

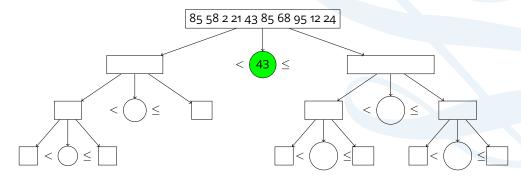
algorithm

Quicksort
Divide & Conque

Comparing

Doson

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Introduction Bubblesort

Selection sort

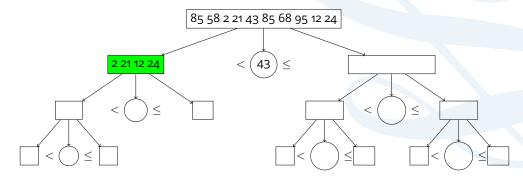
Other

Quicksort
Divide & Conque

. Comparing

Recan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.

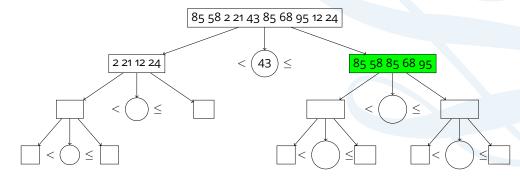




Ouicksort

Select a value from the sequence, this is the pivot.

- Put all values < pivot in one group.
- Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Introduction Bubblesort

Selection sort

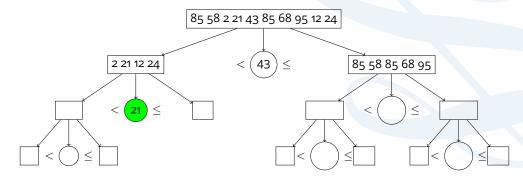
algorithn

Quicksort
Divide & Conque

Comparing

Recan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





Selection sort

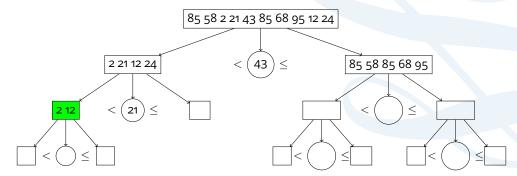
algorithm

Quicksort
Divide & Conque

Comparing

Recan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





Selection sor

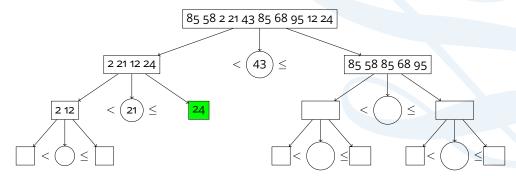
algorithn

Quicksort
Divide & Conque

Comparing

Posan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





Bubblesort
Stable sort

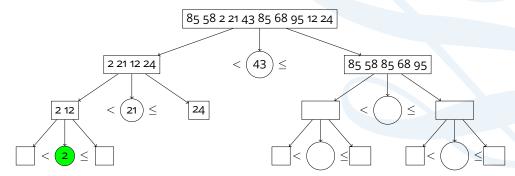
Selection sort

algorithn

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





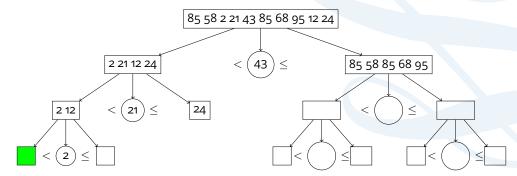
Selection sort

algorithn

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





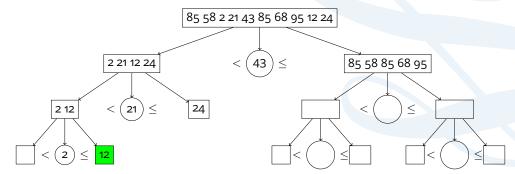
Selection sort

algorithn

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Selection sort

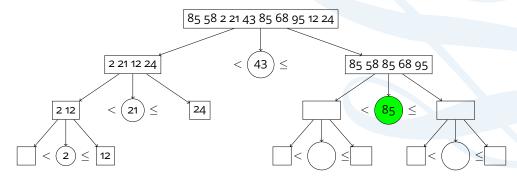
algorithn

Quicksort
Divide & Conque

Comparing

Recan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





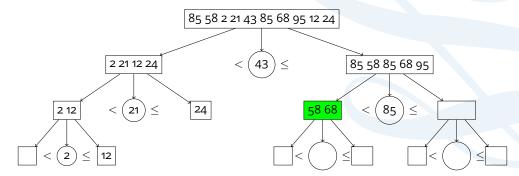
Selection sor

algorithn

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





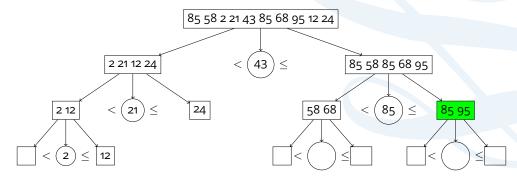
Selection sort

algorithm

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





Selection sort

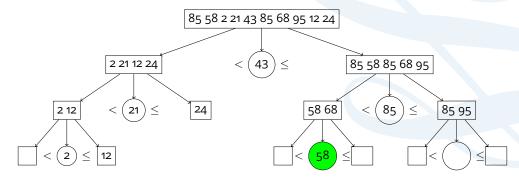
algorithn

Quicksort
Divide & Conque

Comparing

Recan

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





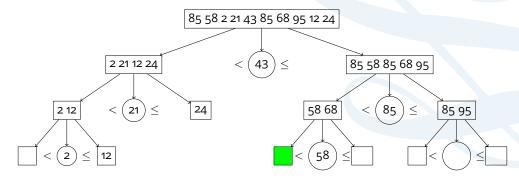
Selection sort

algorithn

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





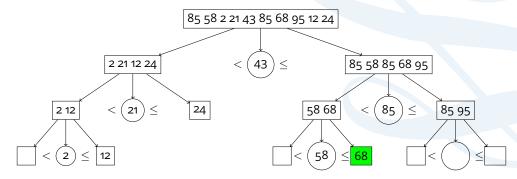
Selection sort

algorithm

Quicksort
Divide & Conque

Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- 4 Treat each group as a new sequence and repeat from step 1.





Selection sort

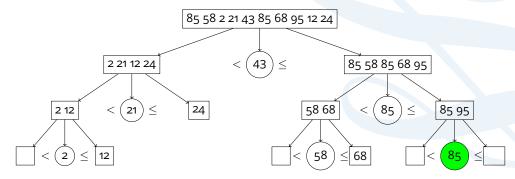
algorithn

Quicksort
Divide & Conque

. Comparing

Danne

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.







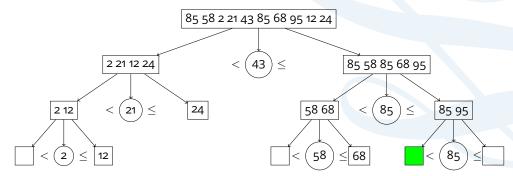
Selection sort

algorithn

Quicksort
Divide & Conque

. Comparing

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Selection sort

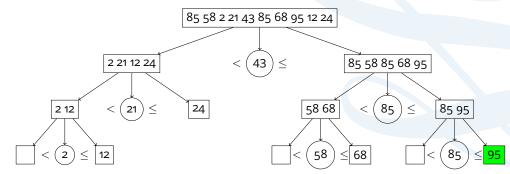
algorithm

Quicksort
Divide & Conque

Comparing

Posan

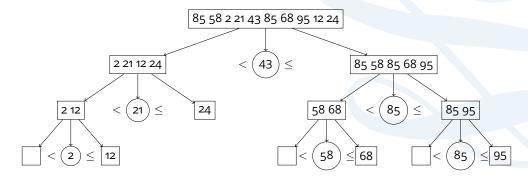
- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.</p>
- 3 Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.





Ouicksort

- Select a value from the sequence, this is the pivot.
- Put all values < pivot in one group.
- Put all values > pivot in another group.
- Treat each group as a new sequence and repeat from step 1.



Selection sor

algorith

Quicksort
Divide & Conqu

Comparing

кеса

Quicksort is...

- ...sometimes in-place.
 - Depends on implementation.
- ...sometimes stable.
 - Depends on implementation.

Some issues with the original algorithms (1959).

- Choosing the pivot.
 - First element.
 - Middle element.
 - Average of first, middle and last.
- Repeated elements.
 - Fat partition.



Introduction

Bubblesor Stable sort In-place

Selection sort

Other algorithms

Divide & Conque

Comparing

Recap

Quicksort is a divide and conquer algorithm.

- Too hard to sort the whole sequence?
- Divide the problem.
 - Still too hard?
 - Divide the problem.
 - Still too hard?
 - Divide the problem.
 - Etc, etc, etc.

Naturally suited for parallelism.



Introduction

Bubblesort

Soloction cort

Other

algorithms

Comparing

Pocan

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.



Comparing

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?



Comparing

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?
- What are you sorting?
 - Linked lists?
 - Sequential memory (arrays)?



Introductior Bubblesort Stable sort

Selection sor

Other algorithms

Comparing

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?
- What are you sorting?
 - Linked lists?
 - Sequential memory (arrays)?
- Where are you sorting?
 - RAM?
 - EEPROM? cheap to read, expensive to write.



1

Introductior Bubblesort Stable sort

Selection sor

Other algorithms

Comparing

compani

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?
- What are you sorting?
 - Linked lists?
 - Sequential memory (arrays)?
- Where are you sorting?
 - RAM?
 - EEPROM? cheap to read, expensive to write.
- \bigcirc Size of n.
 - Insertion sort with small n.





Introductior Bubblesort Stable sort

Selection sor

Other algorithms

Comparing

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?
- What are you sorting?
 - Linked lists?
 - Sequential memory (arrays)?
- Where are you sorting?
 - RAM?
 - EEPROM? cheap to read, expensive to write.
- Size of *n*.
 - Insertion sort with small *n*.
- Consistent performance.
 - Selection sort.





Introductior Bubblesort Stable sort

Selection sor

Other algorithms

Comparing

- Best sorting algorithm depends on multiple factors.
- Good in one situation is bad in another.
- Stability? In place?
- What are you sorting?
 - Linked lists?
 - Sequential memory (arrays)?
- Where are you sorting?
 - RAM?
 - EEPROM? cheap to read, expensive to write.
- Size of *n*.
 - Insertion sort with small *n*.
- Consistent performance.
 - Selection sort.





Sorting

David Croft

Introduction

Bubblesort

In-place

Selection Sort

otner

Quicksort

Companing

Recap

Quiz



Bubblesort Stable sort In-place

Selection sort

algorithm

Quicksort Divide & Cone

Comparing

Recap

- Many sorting algorithms.
- Bubblesort.
- Selection sort.
- Quicksort
- Advantages/disadvantages.
 - In place.
 - In order.
 - Divide and Conquer.
- Performance
 - O()
 - Sequence type.
 - Read/writes.
 - Size of *n*.



Sorting

David Croft

Introduction

Bubblesort

Selection sort

Other

angorianii

Divide & Conq

Comparing

Recap

The End

