

Algorithm performance



Faster sorting



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

By the end of this video you will be able to...

- Describe the merge sort algorithm
- Explain the use of recursion in merge sort
- Use asymptotic analysis to compare merge sort with selection sort and with insertion sort

Merge Sort: Basic Algorithm

If list has one element, return.

Divide list in half

Sort first half

Sort second half

Merge sorted lists

Merge Sort: Basic Algorithm

If list has one element, return.

Divide list in half

Sort first half

Sort second half

HOW?

Merge sorted lists

Merge Sort: Basic Algorithm

If list has one element, return.

Divide list in half

Sort first half

Sort second half

HOW? Recursion!!!

Merge sorted lists

Merge Sort: Basic Algorithm

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists

5	3	2	4	1
---	---	---	---	---

Merge Sort: Basic Algorithm

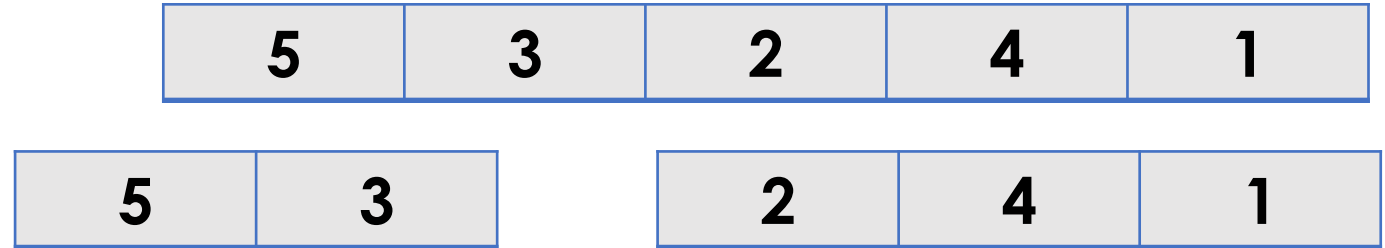
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

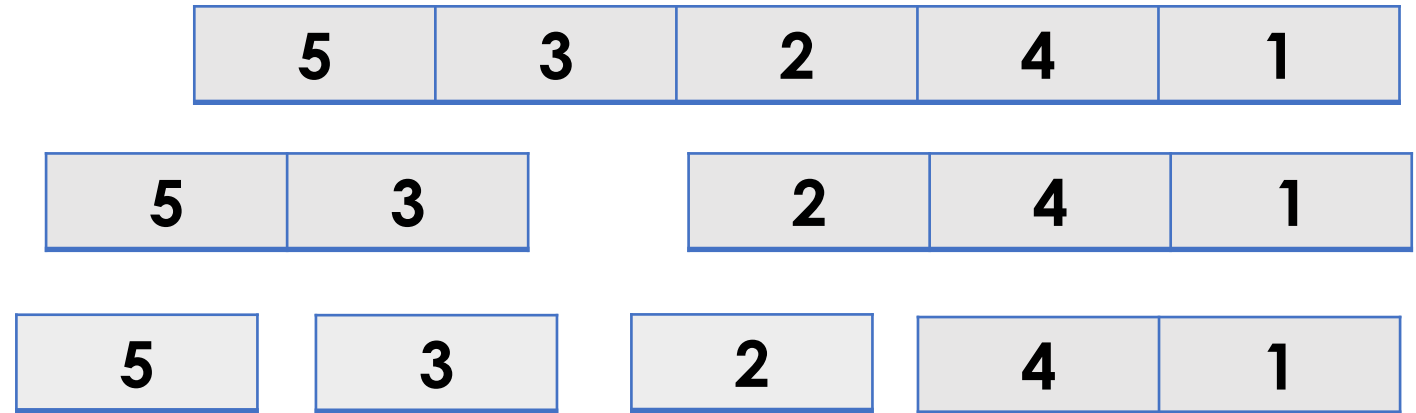
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

If list has one element, return

Divide list in half



Sort first half



Sort second half



Merge sorted lists

Merge Sort: Basic Algorithm

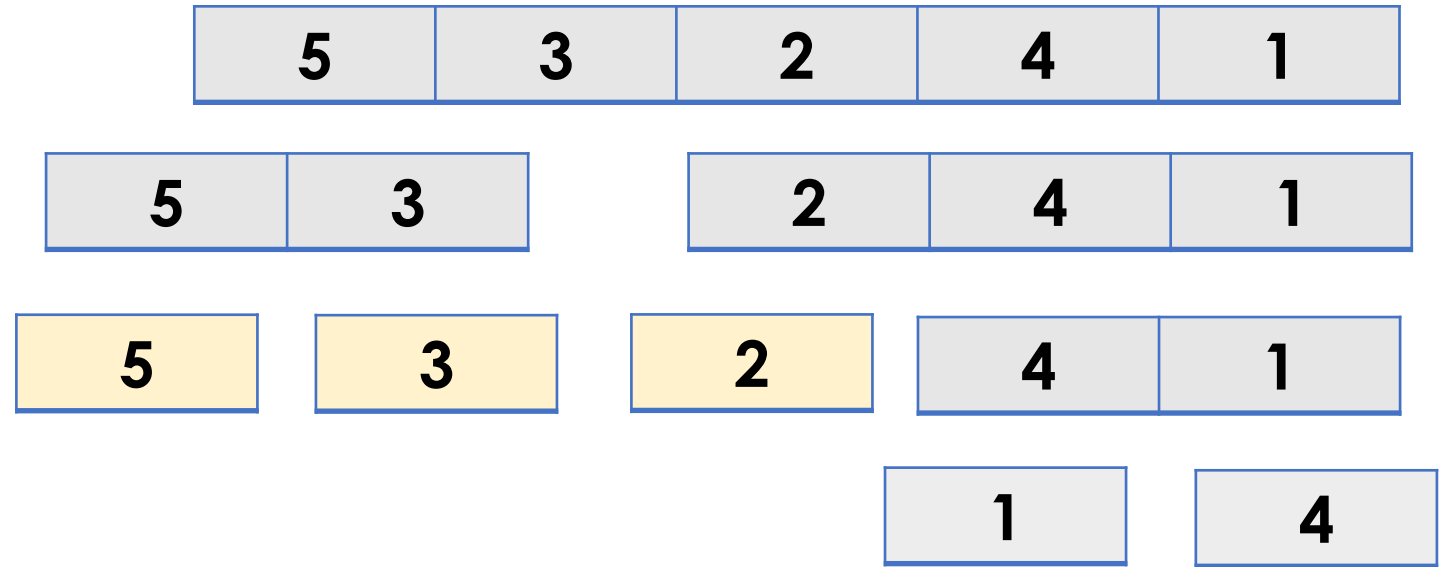
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

If list has one element, return

Divide list in half



Sort first half



Sort second half



Merge sorted lists



Merge Sort: Basic Algorithm

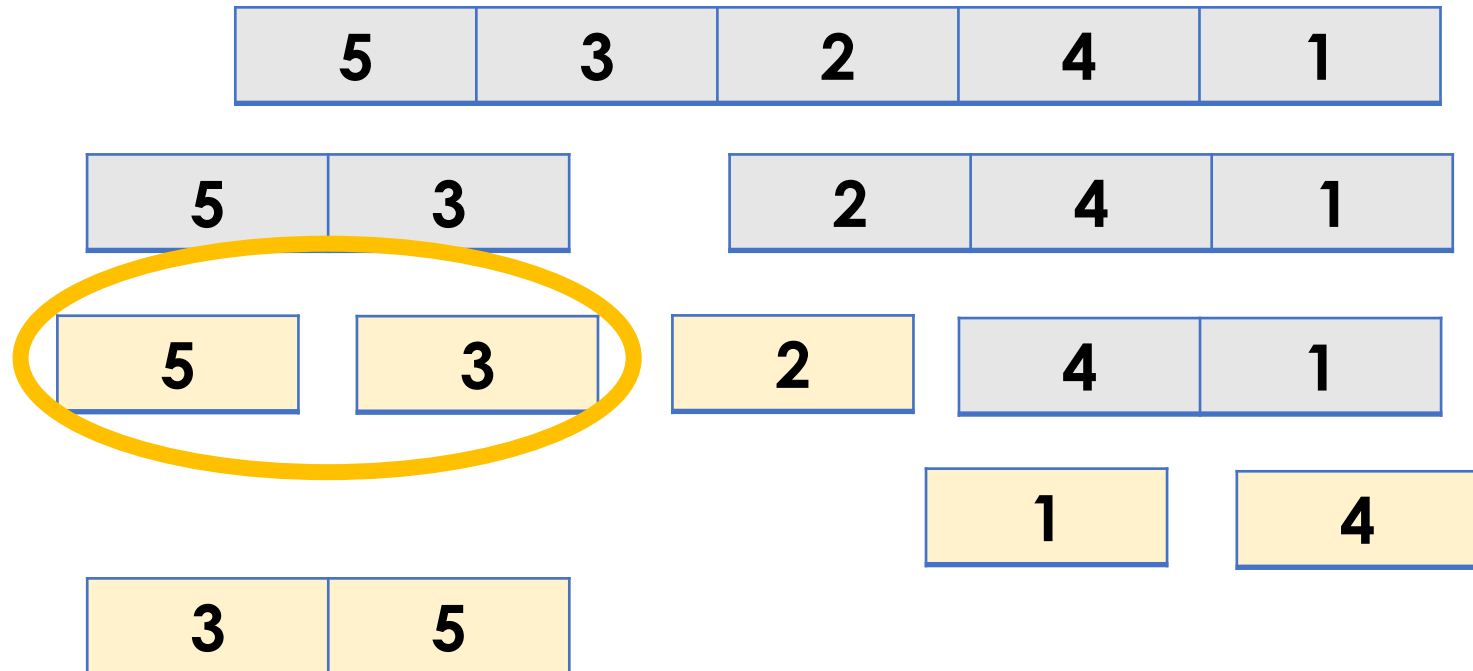
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

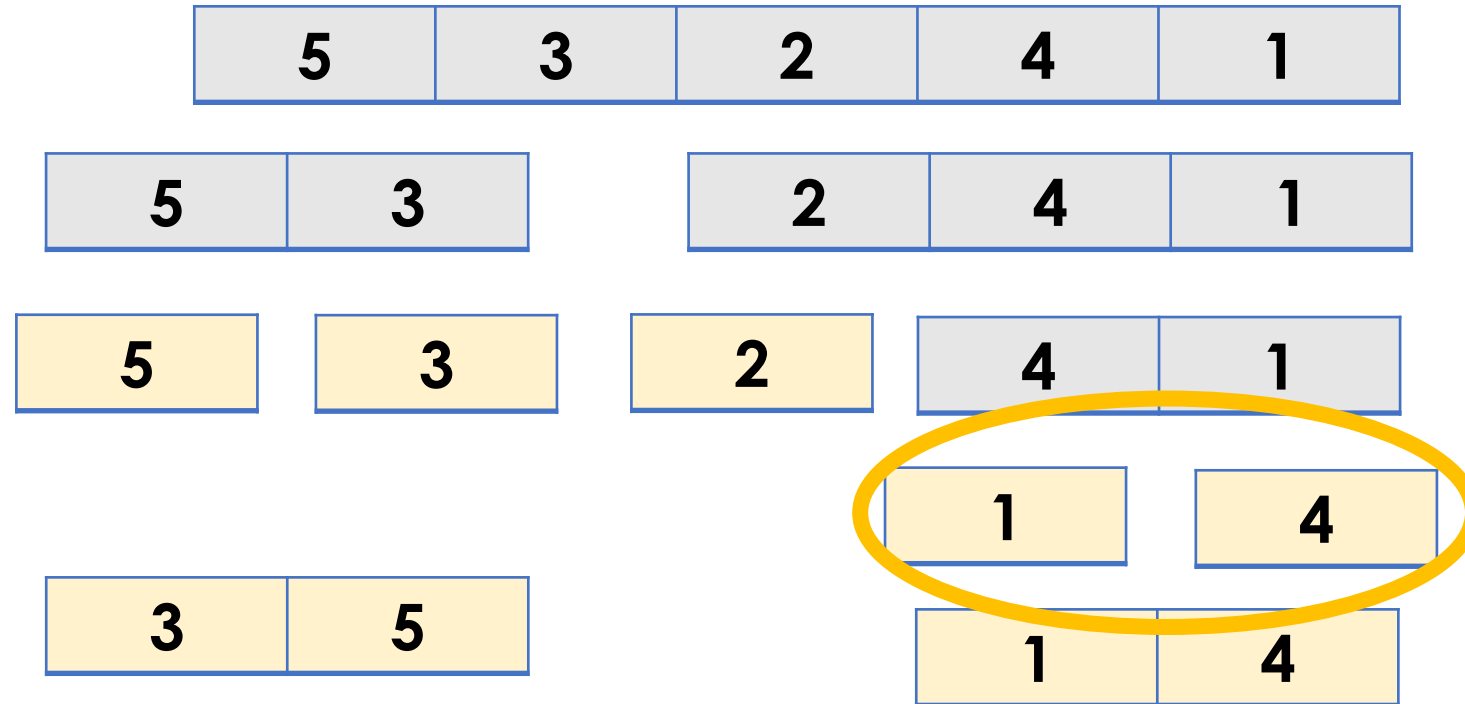
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

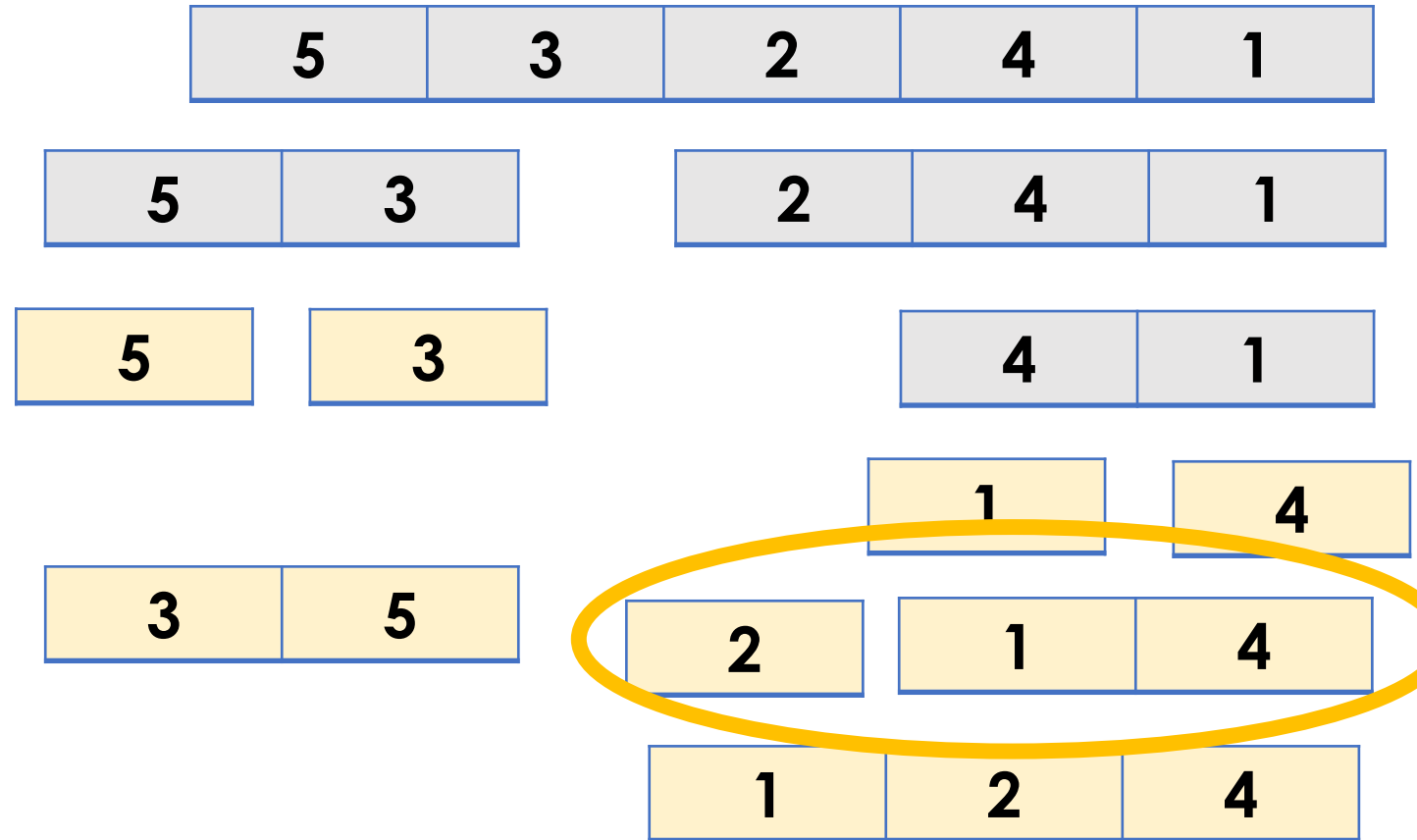
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

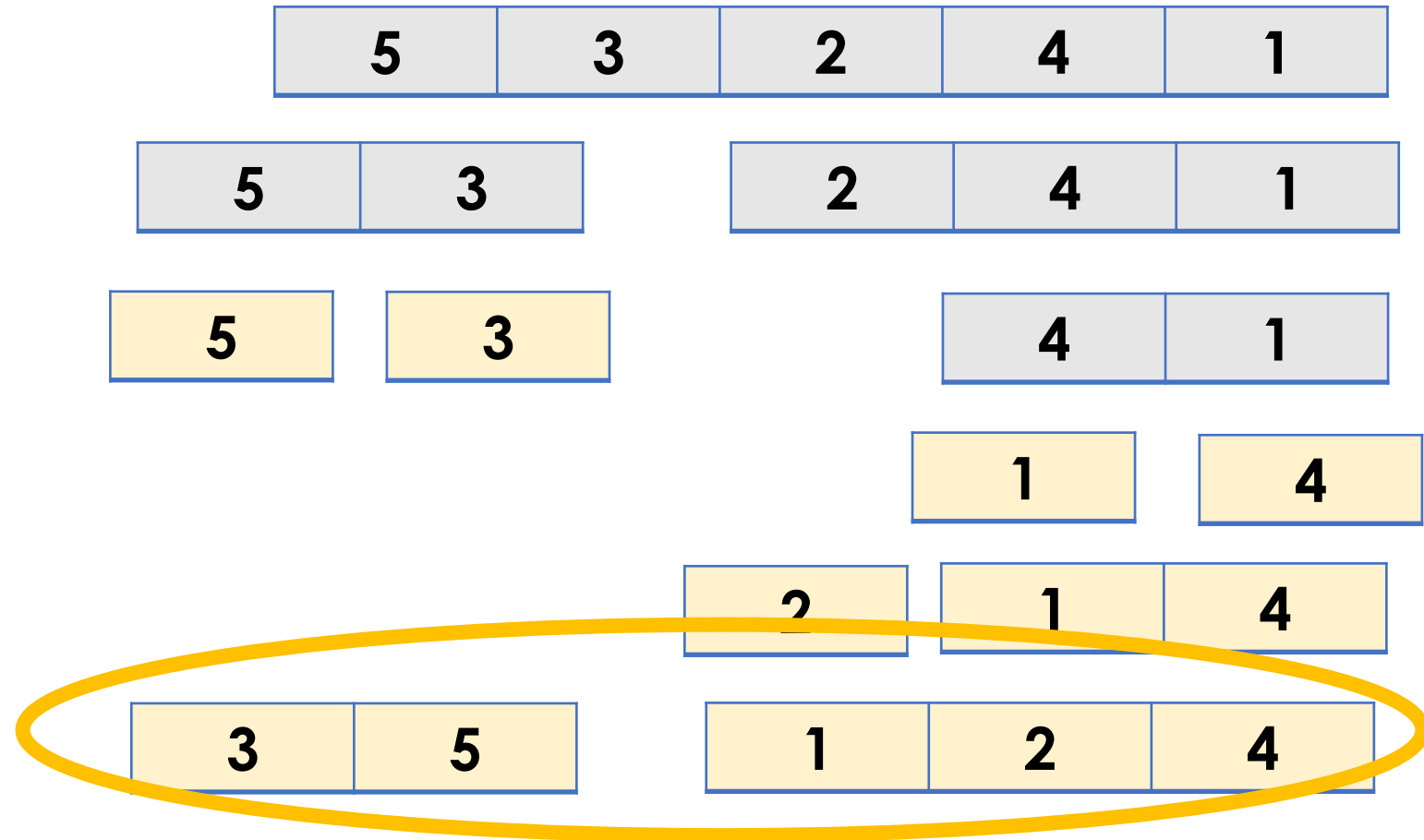
If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



Merge Sort: Basic Algorithm

Performance?

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists

Merge Sort: Basic Algorithm

Performance?

If list has one element, return


Divide list in half

Sort first half

Sort second half

Merge sorted lists

**$O(n)$ work to
merge all the
lists on one level**



Merge Sort: Basic Algorithm

Performance?

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists



**Each time we
divide, we call
MergeSort on
two (smaller)
lists**

Merge Sort: Basic Algorithm

Performance?

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists

**Keep dividing
by two until lists
have size 1**

Merge Sort: Basic Algorithm

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists

Performance?

**Keep dividing
by two until lists
have size 1**

$\log_2(n)$

Merge Sort: Basic Algorithm

Performance?

If list has one element, return

Divide list in half

Sort first half

Sort second half

Merge sorted lists

$O(n \log n)$