



# Merge Sort

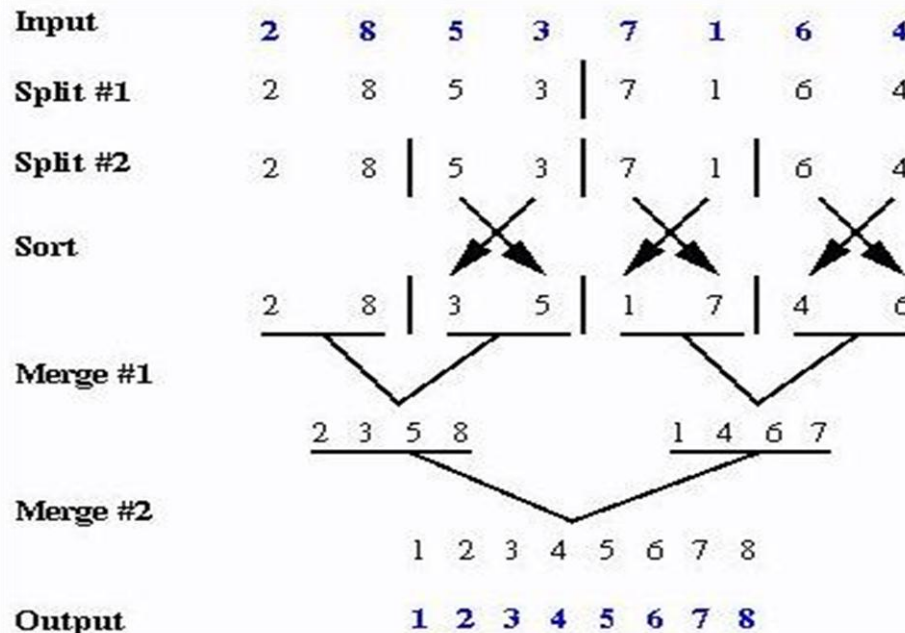
Presented by  
Tyler Adam, Krystina Kamjorn,  
Daniel Nam

CSSE 221

---

# The Basics

- What is a merge sort?
  - Algorithm for sorting data in an array by halving the array, sorting each half, and merging the two halves



# Straightforward Algorithm

- In a sort method, divide the array into two new arrays, each half the original size
- Store the first half of the data in one array and the second half in another
- Recursively call the sort method until the largest split array has one element
- Call a merge method to order and merge the elements in the split arrays
  - Compare the two array elements
  - The smaller element goes into the original array

# Demo

- Merge sorting an array of random numbers
- Public repository → Merge Sort (Section 2)

# Efficiency

- What is merge sort's order?
  - $O(n \log(n))$
- Why?
  - Let's go back and take a look at the merge sort code from the demo.
- Merge sort is more efficient for large arrays
  - For small arrays, merge sort is too slow compared to the amount of code it takes to sort
- "Fun fact" from Wikipedia:
  - For a randomly ordered array and a large  $n$ , merge sort's average number of comparisons approaches  $0.2645 * n$  comparisons fewer than the worst case and the worst case makes 39% fewer comparisons than quick sort makes.

# Comparing Sorts

n	Merge Sort (milliseconds)	Selection Sort (milliseconds)
10,000	40	786
20,000	73	2,148
30,000	134	4,796
40,000	170	9,192
50,000	192	13,321
60,000	205	19,299

# Awesome Sort Animations

- <http://math.hws.edu/TMCM/java/xSortLab/>
- <http://www.sorting-algorithms.com/>
- <http://www.youtube.com/watch?v=qaSqhIMtSVQ>
- [Preliminary Activity](#)
  - <http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/merge/mergen.htm>

# Activity

- On a sheet of paper, write:  
Your Name  
Your Birthday
- Merge sort yourselves according to birthday – January first, December last





Now you know how merge  
sort works!