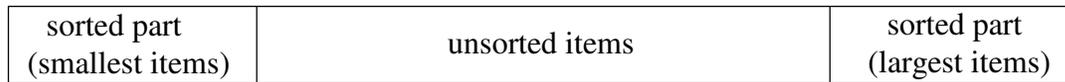


Objective: Become more proficient at implementing sorting algorithms.

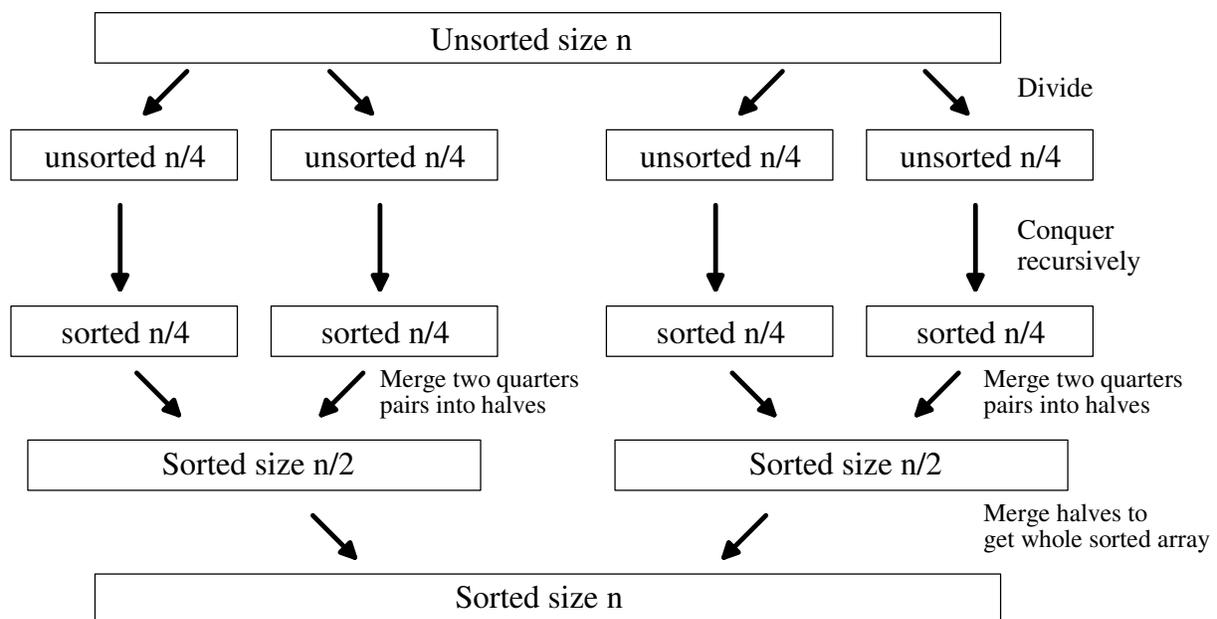
Part A: Implement a modified bubble sort algorithm to “bubble” in both directions. The first pass through the “unsorted” part of the array bubbles the largest item to the right end of the “unsorted” part. The second pass through the “unsorted” part of the array bubbles the smallest item to the left end of the “unsorted” part. This alternating pattern continues until no more passes are necessary.



Include a timing program as in lab8.zip for your program that times the sorting of a randomly generated array of size 10,000 items.

Part B: In class we discussed 2-way merge sort, for this assignment I want you to implement a 4-way merge sort. The general idea of **4-way merge sort** is as follows. Assume “n” items to sort.

- Divide the unsorted part into quarters to get four smaller sorting problems of about equal size = $n/4$
- Conquer/Solve the smaller problems recursively using 4-way merge sort
- “Merge” the solution to the smaller problems together using **two levels of merging** as shown below



NOTE: Use the same `merge` code as used as in the 2-way merger sort code given in lab 8. Just call the 2-way `merge` three times as shown in the above diagram to merge the four quarters. You do not need to rewrite the `merge` code.

Include a timing program as in lab8.zip for your program that times the sorting of a randomly generated array of size 400,000 items.

SUBMISSION

Submit **ALL necessary files** to run your sorts as a single zipped file (called hw5.zip) electronically at

https://www.cs.uni.edu/~schafer/submit/which_course.cgi