

- Problem 1. Recall the Quick sort algorithm partition routine that we used in class. You're told that the following array has just been partitioned around some pivot, $[3, 1, 2, 4, 5, 8, 7, 6, 9]$. Which of the elements could have been the pivot element? (List all that apply; there could be more than one possibility.)
- Problem 2. Design a comparison-based algorithm for sorting a five-element array with seven comparisons in the worst-case.
- Problem 3. Draw decision trees for bubble sort and modified bubble sort on three elements a, b, c , which start in positions indexed by 1, 2, 3 of an array, respectively. Note that bubble sort is inefficient, so it does some redundant comparisons and some nodes will have only one child. What is the average number of comparisons in each case?