

Due Friday, March 17th, 4:00 pm in 2131 Kemper

1. (4 points, 1 point each) Given the values 10, 3, 1, 4, 8, 5, 11, 9, 2, 6, 7 stored in an array in that order, provide the “states” of the array while being sorted for the sorting algorithms specified. Please write the contents of the array on a separate line for each state.
 - a. For insertion sort, show the first 4 states of the array as it is sorted by insertion sort, one state on each line. The “states” are achieved at the end of the for-loop when an entry has been moved to its currently proper position.
 - b. Apply Shellsort using Shell’s increment showing the 3 states of the array. The “states” are achieved at the end of the sorting based on each of the 3 increments.
 - c. Apply heapsort to the array until the 10 and 11 are in the correct positions. Show the 5 states of the heap as you use build heap when each sub-heap has been made into a heap, and then the two states of the array necessary until the 10 is in the proper position.
 - d. Show the first 3 states that occur when applying quicksort to the array using median of 3, and cut-off of 2. A state occurs when quicksort has been applied to a subarray. There should be only one new pivot on each line. Underline the pivot used for each line. Apply the cut-off sort on the next line. When there are multiple subarrays to be sorted, choose the subarray with the smaller values first.
2. (3 points, 1 point each) Weiss 7.20 Using the quicksort implementation in this chapter, (using median of three to determine the pivot), determine the running time of quicksort for
 - a. Sorted input
 - b. Reverse-ordered input
 - c. Random input.
3. (3 points, 1 point each) Weiss 7.21 Repeat Exercise 7.20 when the pivot is chosen as
 - a. The first element
 - b. The larger of the first two distinct elements
 - c. A random elementThis will involve three answers for each part.
4. (4 points, 1 point each) Weiss 7.32 Suppose you are given a sorted list of N element followed by $f(N)$ randomly ordered elements. How would you sort the entire list if
 - a. $f(N) = O(1)$?
 - b. $f(N) = O(\log N)$?
 - c. $f(N) = O(\sqrt{N})$?
 - d. How large can $f(N)$ be for the entire list still to be sortable in $O(N)$ time? Provide the algorithm for your solution.
5. (2 points, 1 point each) Weiss 7.53 “We are given an array that contains N numbers. We want to determine if there are two numbers whose sum equals a given number K . For instance, if the input is 8, 4, 1, 6, and K is 10, then the answer is yes (4 and 6). A number may be used twice.
 - a. Give an $O(N^2)$ algorithm to solve this problem.
 - b. Give an $O(N \log N)$ algorithm to solve this problem. (*Hint*: Sort the items first. After that is done, you can solve the problem in linear time.)