

CS 2420 Section 2 and Section 3
Spring 2016
Assignment 7: Sorting Algorithms
Due: 11:59 p.m. April 9 (Saturday), 2016
Total Points: 100 points

Part I [30 points; 5 points per question]: Submit your solution via Canvas.

Given an array `int A[5] = {17, 23, 5, 11, 2}`, individually apply the following sorting algorithms to sort the data in the **descending order.**

1. Illustrate the results after each pass of the bubble sort. Hint: At most four passes are needed. (refer to slides 7-9 of the class notes)
2. Illustrate the results after each pass of the selection sort. Hint: Exactly four passes are needed. (refer to slides 13-15 of the class notes)
3. Illustrate the results after each pass of the insertion sort. Hint: Exactly four passes are needed. (refer to slides 20-21 of the class notes)
4. Illustrate the results after each pass of the heap sort. Hint: Exactly four passes (e.g., four DeleteMin operations) are needed. For the DeleteMin operation, please show the results of the following two steps: 1) swap the first element with the last element and 2) downheap operation result. (refer to slides 25-42 of the class notes)
5. Illustrate the intermediate results of the merge sort using the tree structure as shown in class. Please label the calling sequence using a set of sequential numbers starting with 1. (refer to slide 62 of the class notes)
6. Illustrate the intermediate results of the quick sort using the tree structure as shown in class. Please label the calling sequence using a set of sequential numbers starting with 1. **You should always choose the last element in the array as the pivot.** (refer to slide 74 of the class notes)

Part II [70 points; 20 points per sorting algorithm; 10 points for other functions/operations]:

Write a C++ program to implement the heap sort algorithm, the merge sort algorithm, and the quick sort algorithm introduced in class. Please use the “starter” cpp file called “Assign7StarterCode.cpp” to add your C++ implementation for each function. You may need to add other necessary functions.

The program first reads the number of elements in the input file “Assign7Data.txt” and then reads the numbers and stores them in the dynamic array A.

- Perform the heap sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.
- Read the numbers from the input file “Assign7Data.txt” and store them in the same dynamic array A. Perform the merge sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.
- Read the numbers from the input file “Assign7Data.txt” and store them in the same dynamic array A. Perform the quick sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.

The program then reads the number of elements in another input file “dictionary.txt”, which was used in Assignment 3, and converts each word into an integer using the strategy mentioned in Assignment 5. That is, go through the string a character at a time, convert each character to an integer (a.k.a., ASCII value), and add up the integer values to convert the key to a nonnegative integer. The converted integer is then stored in the dynamic array A.

- Perform the heap sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.
- Read the strings from the input file “dictionary.txt” and store the converted string in the same dynamic array A. Perform the merge sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.
- Read the strings from the input file “dictionary.txt” and store the converted string in the same dynamic array A. Perform the quick sort algorithm to sort the numbers in the array in the ascending order. After the numbers in the array are sorted, the numbers will be output on the console/screen.

Add appropriate code in the starter code to measure the time of each algorithm to sort the data in “Assign7Data.txt” and the data in “dictionary.txt”, respectively. Summarize your findings on the console/screen.