Seoul National University

M1522.000900 Data Structure

Fall 2019, Kang

Homework 6: Internal Sorting (Chapter 7)

Due: November 18, 11:59 PM

Reminders

- The points of this homework add up to 100.
- Like all homeworks, this has to be done individually.
- Lead T.A.: Seungcheol Park (ant6si@snu.ac.kr)
- Type your answers in English.
- If you have a question about the assignment, upload your question in eTL.
- If you want to use slipdays or consider late submission with penalties, please note that you are allowed four days to submit your assignment after the due date.

Important

- Write your name and student ID inside the document you are submitting. (10% deducted from total score)
- The submitted document must be a pdf. (10% deducted from total score)
- The submitted document should not contain any variation of a scanned document or handwriting, including diagrams or equations hand-drawn from a tablet or computer. (10% deducted from total score)

Remember that:

1. Whenever you are making an assumption, please state it clearly

Question 1

Figure 1 shows a sorting algorithm implementation. Answer the following questions [20points].

```
public static int[] sort(int[] array){
    int[] aux = new int[array.length];
    int min = array[0];
    int max = array[0];
    for (int i = 1; i < array.length; i++){
        if (array[i] < min){
          min = array[i];
        }else if (array[i] > max){
          max = array[i];
        }
    }
    int[] counts = new int[max - min + 1];
    for (int i=0; i<counts.length; i++) {
        counts[i] = 0;
    }
    for (int i = 0;  i < array.length; i++){
        counts[array[i] - min]++;
    }
    counts[0]--;
    for (int i = 1; i < counts.length; i++){
        counts[i] = counts[i] + counts[i-1];
    }
    ★
    for (int i = array.length - 1; i >= 0; i--){
        aux[counts[array[i] - min]--] = array[i];
    }

    return aux;
}
```

Figure 1. A sorting algorithm implementation

(1) If the input array is [6, 3, 2, 5, 1, 0, 5, 3, 4], what is the values of the `counts` array at the star point (★)? [5points]

(2) Let the size of the input array be *n* and the value of *(max-min)* be *k*. What is the time complexity of the algorithm? [5points]

(3) What is the space complexity of the algorithm? [5points]

(4) A sorting algorithm is said to be stable if the original ordering for duplicate keys is preserved. Is this algorithm stable? [5points]

Question 2

Figure 2 shows an optimized merge sort implementation to sort an array of integers in Java. If there are $n = 2^m - 1$ calls to the merge sort function (mergeSort) to sort an array, how many calls will there be to insertion sort (insertionSort)? [20 points]

```
public void mergeSort(int[] a, int[] temp, int l, int r) {
    int mid = (l + r) / 2;
    if (l == r) return;
    // sort the left part
    if (mid - l > Threshold) mergeSort(a, temp, l, mid);
    else insertionSort(a, l, mid);
    // sort the right part
    if (r - mid > Threshold) mergeSort(a, temp, mid + 1, r);
    else insertionSort(a, mid + 1, r);
    // merge
    for (int i = l; i <= mid; i++) temp[i] = a[i];
    for (int i = 1; i <= r - mid; i++) temp[r - i + 1] = a[mid + i];
    for (int i = l, j = r, k = l; k <= r; k++) {
        if (temp[i] < temp[j]) {
            a[k] = temp[i]; i++;
        } else {
            a[k] = temp[j]; j--;
        }
    }
}
```

Figure 2. An optimized merge sort implementation in Java.

Question 3

`QuickSort` solves the problem by dividing the list into two sub-lists recursively. Assume that the ratio of the two sub-lists is always 1:99. What is the time complexity of sorting? Justify your answer [20points].

Question 4

Although Radix sort works only for sorting numbers, it actually can be applied to other compatible datatype, for example, English letters. The order of English letters follows alphabetic order.
Sort following list of English words using Radix sort:
COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX
Radix sort requires 3 passes for sorting these words. What is the result of the 2nd pass? [20 points]

Question 5

Let A[1,2, … ,n] be an array of n distinct numbers. If i< j and A[i] > A[j], then the pair (i, j) is called an *inversion* of A. [20 points]

(1) List the five inversions of the array (2, 3, 8, 6, 1) [5 points]

(2) Which array with elements from the set {1, 2, 3, …, n} has the largest number of inversions? How many inversions does it have? [5 points]

(3) What is the relationship between the number of comparisons of insertion sort and the number of inversions in the input array? Justify your answer. [10 points]