CX2101 Algorithm Design and Analysis **2020/21 Semester 2**

Tutorial 1: Sorting

School of Computer Science and Engineering *Nanyang Technological University*

Week 3 (Q1-Q3):

Q1 The worst case for Insertion Sort occurs when the keys are initially in decreasing order. Suppose the performance of Insertion Sort is measured by the total number of comparisons between array elements (not counting the number of swaps). Show at least two other initial arrangements of keys that are also worst cases for Insertion Sort.

Q2 Use the divide and conquer approach to design an algorithm that finds both the largest and the smallest elements in an array of n integers. Show that your algorithm does at most roughly $1.5n$ comparisons of the elements. Assume $n = 2^k$.

Q3 Show how Merge Sort sorts each of the arrays below and give the number of comparisons among array elements in sorting each array.

(a) 14 40 31 28 3 15 17 51

(b) 23 23 23 23 23 23 23 23

Week 4 (Q4-Q6):

Q4 Suppose that, instead of using E[middle] as pivot, Quick Sort also can use the median of E[first], E[(first + last)/2] and E[last]. How many key comparisons will Quick Sort do in the worst case to sort n elements? (Remember to count the comparisons done in choosing the pivot.)

Q5 Each of n elements in an array may have one of the key values red, white, or blue. Give an efficient algorithm for rearranging the elements so that all the reds come before all the whites, and all the whites come before all the blues. (It may happen that there are no elements of one or two of the colours.) The only operations permitted on the elements are examination of a key to find out what colour it is, and a swap, or interchange, of two elements (specified by their indices). What is the asymptotic order of the worst case running time of your algorithm? (There is a linear-time solution.)

Q6 Suppose we have an unsorted array A of n elements and we want to know if the array contains any duplicate elements.

(a) Outline (clearly) an efficient method for solving this problem.

(b) What is the asymptotic order of the running time of your method in the worst case? Justify your answer.

(c) Suppose we know the n elements are integers from the range $1, \ldots, 2n$, so other operations besides comparing keys may be done. Give an algorithm for the same problem that is specialized to use this information. Tell the asymptotic order of the worst case running time for this solution. It should be of lower order than your solution for part (a).

Week 5 (Q7-Q9):

Q7 Given an array with content (of type date): 1 Jul, 30 Jan, 22 Mar, 22 Dec, 30 May, 21 Feb, 3 Nov, 7 Jun, 22 Feb, 21 Nov, 30 Dec; all of the same year. Suppose an earlier date is considered bigger than a later date; for example, "30 Jan is bigger than "30 Dec. In order to sort these dates by Heap Sort, let us construct a maximizing heap. Show the content of the array after the heap construction phase.

Q8 An array of distinct keys in decreasing order is to be sorted (into increasing order) by Heap Sort.

(a) How many comparisons of keys are done in the heap construction phase (Algorithm constructHeap() in lecture notes on Sorting) if there are 10 elements?

(b) How many are done if there are n elements? Show how you derive your answer.

(c) Is an array in decreasing order the best case, the worst case, or an intermediate case for this algorithm? Justify your answer.

Q9 Given k lists with a total of n numbers, where $k \geq 2$ and each list has been sorted in decreasing order, design an algorithm to merge the k lists into one list sorted in decreasing order, in running time $\mathcal{O}(n \log_2 k)$.