

# EL9343 Homework 5

Due: Oct. 13th 11:00 a.m.

1. We can improve the running time of quicksort in practice by taking advantage of the fast running time of insertion sort when its input is "nearly" sorted. Upon calling quicksort on a subarray with fewer than  $k$  elements, let it simply return without sorting the subarray. After the top-level call to quicksort returns, run insertion sort on the entire array to finish the sorting process. Argue that this sorting algorithm runs in  $O(nk + n \lg(\frac{n}{k}))$  expected time. How should we pick  $k$ , both in theory and in practice?
2. **Quicksort with equal element values.** The analysis of the expected running time of the randomized quicksort assumes that all element values are distinct. In this problem, we examine what happens when they are not.
  - (a) Suppose that all element values are equal. What would be randomized quicksort's running time in this case?
  - (b) The PARTITION procedure returns an index  $q$  such that each element of  $A[p, \dots, q-1]$  is less than or equal to  $A[q]$  and each element of  $A[q+1, \dots, r]$  is greater than  $A[q]$ . Modify the PARTITION procedure to produce a procedure PARTITION'( $A, p, r$ ), which permutes the elements of  $A[p, \dots, r]$  and returns two indices  $q$  and  $t$ , where  $p \leq q \leq t \leq r$ , such that
    - i. all elements of  $A[q, \dots, t]$  are equal,
    - ii. each element of  $A[p, \dots, q-1]$  is less than  $A[q]$ , and
    - iii. each element of  $A[t+1, \dots, r]$  is greater than  $A[q]$ .Like PARTITION, your PARTITION' procedure should take  $\Theta(r-p)$  time.
  - (c) Modify the RANDOMIZED-PARTITION procedure to call PARTITION', and name the new procedure RANDOMIZED-PARTITION'. Then modify the QUICKSORT procedure to produce a procedure QUICKSORT'( $A, p, r$ ) that calls RANDOMIZED-PARTITION' and recurses only on partitions of elements not known to be equal to each other.
  - (d) Using QUICKSORT', how would you adjust the analysis of randomized quicksort to avoid the assumption that all elements are distinct?
3. Similar to the example shown in slides, illustrate the operation of COUNTING-SORT on

$$A = [5, 9, 7, 2, 3, 6, 9, 5, 2, 1]$$

4. Prove the lower bound of  $\lceil \frac{3n}{2} \rceil - 2$  comparisons in the worst case to find both the maximum and minimum of  $n$  numbers. (Hint: consider how many numbers are potentially either the maximum or minimum, and investigate how a comparison affects these counts.)
5. **Largest  $i$  numbers in sorted order.** Given a set of  $n$  numbers, we wish to find the  $i$  largest in sorted order using a comparison-based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst-case running time, and analyze the running times of the algorithms in terms of  $n$  and  $i$ .
  - (a) Sort the numbers, and list the  $i$  largest.
  - (b) Build a max-priority queue from the numbers, and call EXTRACT-MAX  $i$  times.
  - (c) Use an order-statistic algorithm to find the  $i$ th largest number, partition around that number, and sort the  $i$  largest numbers.