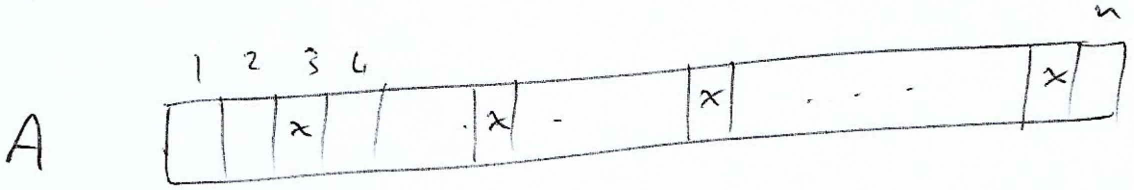
8.1-1

what is the smallest number of comparisons required to be certain. that an array is sorted?

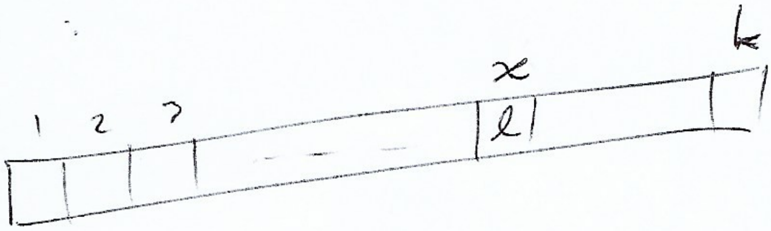
Should be $n-1$.

8.2-2

Suppose A has multiple occurrences of some number, say x occurs l times $(x \leq k)$



Then in C we'll have l in position x



Because the last loop of Counting Sort starts at the right of the array, the first x to be positioned will be the one furthest right.

Then the next x encountered from the right will be placed to the left of the first, thus maintaining the order of those two x's.

For the remaining x's the same reasoning holds.

8.2 - 4

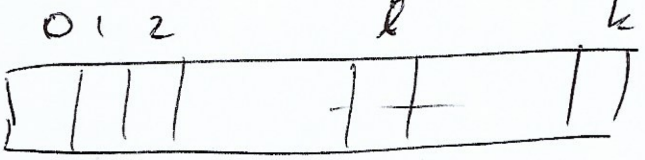
Given A

where each entry is in range 0 → k.

Do lines 1 → 8 of Counting Sort.

This creates a new array:

C



where the value at C[l] is the number of entries in A that are less or equal to l.

Given a and b in range 0 → k, the number of values in A that fall into the range [a ... b] is the

C[b] - C[a-1].

This step is done in O(1) time.

Constructing C is done in O(n+k) time.

8.3 - 2

InsertionSort & Mergesort are Stable.

You can check that if some value occurs more than once, those values will not be swapped during sorting

Quicksort is not stable.

You can show this with a small example, such as $\boxed{7 \mid 7 \mid 7}$

One way to make any sort stable is to add an array I of labels that records the order of occurrences.

E.g.

A	7	8	7	10	3	8	7	3	4	8	7
I	1	1	2	1	1	2	3	2	1	3	4

Then use a lexicographic order when sorting: $\begin{array}{cc} A & \boxed{x} \\ I & \boxed{i} \end{array} \leq \begin{array}{c} \boxed{y} \\ \boxed{j} \end{array}$ if either $x < y$

or $x = y$ & $i < j$

8.3 - 4

Given array A of length n where each entry is an integer in range 0 → n^3-1.

Convert each entry into a base n number. So each entry in A will be a 3-digit number: $a_2a_1a_0$ where each a_i is in range 0 → n-1

Now apply Radix Sort: there are 3 columns and the range of values in each column is 0 → n, so the run-time is $\Theta(3(n+n)) = \Theta(n)$

8.4 - 2

Use an $O(n \log n)$ sorting algorithm to sort each bucket, instead of Insertion-sort.