Algorithms and Data Structures 2020/21
Week 6 tutorial sheet

1. Draw the decision tree (under the assumption of all-distinct inputs) for QUICKSORT for $n = 3$.

2. What is the smallest possible depth of a leaf in a decision tree for a sorting algorithm?

   This is Ex 8.1-1 of [CLRS] (2nd and 3rd ed).

3. Consider the task of sorting just 4 numbers. We consider how many comparisons are needed to do this, where a comparison returns either $x \leq y$ or $x > y$, for any pair x, y.

   (a) Give an algorithm which sorts 4 numbers using 5 comparisons in the worst-case.

   (b) Use the decision-based model from Lecture 7 to *prove* that *any algorithm* to sort 4 numbers will need to use 5 comparisons in the worst case.

4. A sorting algorithm is said to be *stable* if for every pair of indices $i < j$ such that $A[i] = A[j]$ in the input array, the sorting algorithm places the element $A[i]$ before $A[j]$ in the sorted output (i.e., we only exchange the *relative* position of items if we *need* to).

   Quicksort is not stable. Can you come up with an input array with at most 2 duplicates of any number, on which quicksort is not stable? (Try to get this to stay true during the recursive levels also).

5. During the running of the procedure QUICKSORT, how many times do we consider a pivot in the worst case? How does the answer change in the best case?

6. Show how to sort n integers in the range $\{1, \ldots, n^2\}$ in $O(n)$ time.

   Note: This depends on the material in Lecture 9.