Tutorial (week 10)

All the answers can be given with pseudocode or with Python. Remember that most of the time, many solutions are possible.

1. Consider the following list $L = [54, 19, 93, 17, 77, 31, 44, 55, 20]$. when applying a recursive quick-sort algorithm on L, explain with a binary tree the successive recursive calls and with another binary tree the successive merge processes.

2. Show that the best-case running time of quick-sort on a sequence of size n with distinct elements is $O(n \log n)$.

3. Suppose we modify the deterministic version of the quick-sort algorithm so that, instead of selecting the last element in an n-element sequence as the pivot, we choose the element at index $\lfloor n/2 \rfloor$, that is, an element in the middle of the sequence. What is the running time of this version of quick-sort on a sequence that is already sorted ?

4. Consider again the modification of the deterministic version of the quick-sort algorithm so that, instead of selecting the last element in an n-element sequence as the pivot, we choose the element at index $\lfloor n/2 \rfloor$. Describe the kind of sequence that would cause this version of quick-sort to run in $\Theta(n^2)$ time.

5. Suppose we are given a sequence S of n elements, on which a total order relation is defined. Describe an efficient method for determining whether there are two equal elements in S. What is the running time of your method ?

6. Let A and B be two sequences of n integers each. Given an integer x, describe an $O(n \log n)$-time algorithm for determining if there is an integer a in A and an integer b in B such that $x = a + b$.

7. Given a sequence of numbers, $(x_1, x_2, ..., x_n)$, the mode is the value that appears the most number of times in this sequence. Give an efficient algorithm to compute the mode for a sequence of n numbers. What is the running time of your method ?

8. Develop an algorithm that computes the k-th smallest element in a set of n distinct integers with an average complexity $O(n)$ time.

Hints

- **Question 3: Consider how the pivots work on a sorted sequence.**
- **Question 4: Review the worst-case performance example for standard quick-sort.**
- **Question 5: Sort first.**
- **Question 8: Consider a strategy similar to the quicksort (using array partition according to a pivot value).**