4.1. Given an array of n numbers from 1 to k, develop a data structure that can answer queries like «How many elements in the array are in range from a to b?» in $O(1)$ time. Time for pre-calculation $O(n+k)$.

4.2. How to sort strings (for example, consisting only of Latin letters) in lexicographic order using radix sort in $O(\sum len(s_i)$?

4.3. There is an array of n non-negative integers. Find the smallest integer that is not in the array in $O(n)$ time.

4.4. There is an array of $2n$ different elements on which a linear order is defined (that is, any two elements can be compared). You need to split them into n pairs, so that the segments whose boundaries are numbers from pairs do not intersect (for example, from array $(4, 10, 1, 6, 7, 2)$ you can build pairs $(1, 2)$, $(7, 10)$, $(4, 6)$). Is it possible to solve this problem faster than in $O(n \log n)$ time?

4.5. Build the sorting network for the bubble sort algorithm. What is the number of comparators and what is the depth of this network?

4.6. There are n friends living on the line, the i-th friend lives at x_i. They want to meet at one point. Help them find a point so that the total distance they travel is minimal.

4.7. There are n friends live on the line, the i-th friend lives at x_i. They want to meet at one point. Help them find a point so that the sum of the squares of the distances they travel is minimal.

4.8. There is an array of n non-negative integers. You can decrease the numbers, but so that they remain non-negative. What is the maximum number of distinct numbers that an array can have after several such operations?