Lower Bounds and Sorting Algorithms

1. There are n stones, and there is the only radioactive stone among them. We use the Geiger counter to find it. By a single check we can test whether there is a radioactive stone in any pile of stones. Find the least number of checks to find the radioactive stone.

2. Alice has written integers from 1 to 64 in the chessboard's cells (each number occurs exactly once). Bob can not see the chessboard, but he can give Allice a list of cells' ids, and she replies with the list of numbers written in the listed cells (disregarding the order). Find the least number of queries that required restoring Alice's enumeration.

3 [Shen 4.5.3-4]. Suppose we have $2n$ objects (say, stones) of different weights and a balance that can be used to find which of any two given stones is heavier.

1. Prove that it is possible to find the stones with minimal and maximal weights among $2n$ stones using only $3n - 2$ comparisons.

2. Prove that no algorithm can find the stones with minimal and maximal weights among $2n$ stones using less than $3n - 2$ comparisons in the worst case.

4 [Shen 4.5.5-6]. 1. Assume that n stones of different weights are given. Find both the stone with maximal weight and the second best using at most $n + \lceil \log_2 n \rceil - 2$ comparisons.

2*. Prove that no algorithm can find both the winner and the second best stone (among n stones of different weights) using less than $n + \lceil \log_2 n \rceil - 2$ comparisons in the worst case.

5. The input of the problem is natural numbers n and $k < n$ and an array of n natural numbers, each of which is at most k. Suggest an asymptotically best algorithm that sorts the array.

6. There are n words of length k, consisting of small letters of the Latin alphabet. Suggest a linear-time algorithm for sorting them in lexicographic (dictionary) order.