Домашнее задание

1. Дано n слов длины k, состоящих из маленьких букв латинского алфавита. Предложите эффективный алгоритм их сортировки в лексикографическом (словарном) порядке.

2. Пусть числовой массив $a[1], \ldots, a[n]$ строго унимодален на максимум. Это означает, что существует t, такое что

$$a[1] < a[2] < \cdots < a[t] > a[t+1] > \cdots > a[n-1] > a[n], \quad 1 \leqslant t \leqslant n.$$

1. Разрешается за один ход спросить значение одного элемента массива. Докажите, что можно найти значение максимального элемента $a[t]$ за не более $O(\log n)$ ходов.

2*. Докажите, что если $f : \mathbb{R} \to \mathbb{R}$ строго выпуклая функция ($\forall x, y \in \mathbb{R}, \forall a \in (0,1)$ выполнено $(1-a) \cdot f(x) + a \cdot f(y) > f((1-a) \cdot x + a \cdot y)$), то она строго унимодальна на минимум (определение для непрерывного случая аналогично дискретному). Также докажите, что если f – дважды дифференцируемая, то из $f''(x) > 0$ следует, что она строго выпуклая.

3*. Дано k точек на плоскости, вещественные координаты которых по модулю не превосходят N. Нужно найти точку x_0 на оси Ox, суммарное расстояние от которой до всех точек минимально. Вам требуется найти эту точку с точностью до ε (т. е. ваш ответ x не должен отличаться от x_0 более, чем на ε по модулю). Предложите алгоритм, который решает эту задачу за $O(k \cdot \log \frac{N}{\varepsilon})$ арифметических операций и операций извлечения корня.

3. Имеется n монет, среди которых одна фальшивая, и чашечные весы. Настоящие монеты все имеют одинаковый вес, а фальшивая легче. На каждую чашку весов можно класть произвольное количество монет. Докажите, что фальшивую монету можно найти за $\log_3 n + c$ взвешиваний.

4. Докажите, что в условиях предыдущей задачи для нахождения фальшивой монеты необходимо $\log_3 n + c$ взвешиваний.

5. Даны два отсортированных массива длины n. Предложите как можно более эффективный алгоритм поиска медианы в массиве, состоящем из всех данных $2n$ элементов. Можно считать, что все элементы различные. Докажите корректность алгоритма и оцените его сложность (количество сравнений). В этой задаче обращения к элементам массива выполняются за O(1), читать оба массива целиком не требуется, считайте, что они уже лежат в памяти.

Определение. Сортировку называют *устойчивой*, если равные по величине элементы в результате сортировки оказались расположены в порядке вхождения в исходный массив. Формально, если $a[i] = a[j], i < j$ и в результате сортировки массива a i-ый элемент оказался на месте i', а j-ый на j', то $i' < j'$.

Определение. Сортировка называется *in-place*, если при её работе используется только память, отведённая под массив и $O(1)$ вспомогательной памяти.

6. Для каждой сортировки из списка определите, является ли она устойчивой и in-place: **а)** QuickSort; **б)** MergeSort; **в)** InsertionSort; **г)** HeapSort.

Формальное определение алгоритмов можно найти в книге Кормена и др.

7. Докажите, что каждую сортировку сравнениями можно сделать устойчивой, сохранив асимптотическое время её работы.

8. Определите, что число является значением данного многочлена с натуральными коэффициентами в натуральной точке. На вход задачи подаются натуральные числа $n, a_0, \ldots a_n$ и y. Необходимо определить, существует ли натуральное число x, такое что

$$y = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0.$$

9. Ваш лектор по алгоритмам нашёл два одинаковых прочных шарика из неизвестного материала и внезапно решил измерить их прочность в этажах стоэтажного небоскрёба: прочность равна номеру минимального этажа, при броске шарика из окна которого шарик разобъётся (максимум 100). Считаем, что если шарик уцелел, то его прочность после броска не уменьшилась. Сколько бросков шариков необходимо и достаточно для нахождения прочности?