Задание 4. Рекурренты, сортировки, нижние оценки.

1 На вход задачи поступают три отсортированных массива. Постройте алгоритм, находящий число уникальных элементов в объединении этих массивов.

2 К серверу приходят одновременно n клиентов. Для клиента i известно время его обслуживания t_i. Время ожидания клиента определяется как сумма времени обслуживания всех предыдущих клиентов и времени обслуживания его самого. К примеру, если обслуживает клиентов в порядке номеров, то время ожидания клиента i будет равно $\sum_{j=1}^{i} t_j$. Постройте эффективный алгоритм, находящий последовательность обслуживания клиентов с минимальным суммарным временем ожидания клиентов, докажите его корректность и оцените асимптотику.

3 На вход задачи поступает массив a из n чисел. Постройте алгоритм, находящий число инверсий в массиве, то есть таких пар индексов i, j, что $i < j$ и $a[i] > a[j]$. **Рекомендация:** модифицируйте алгоритм сортировки слиянием.

4 На вход поступает число n и массив a размера $2n+1$. Постройте алгоритм, находящий число s, минимизирующее сумму $\sum_{i=1}^{2n+1} |a_i - s|$. Докажите корректность алгоритма, просто сослаться на факт из статистики недостаточно.

5 Дан массив из n чисел. Нужно разбить этот массив на максимальное количество непрерывных подмассивов так, чтобы после сортировки элементов внутри каждого подмассива весь массив стал отсортированным. Предложите $O(n \log n)$ алгоритм для решения этой задачи.

6 Найдите асимптотическую оценку функции $T(n)$. Примените мастер-теорему в тех случаях, когда ее можно использовать, **и посчитайте асимптотику иначе, когда нельзя**. Варианты есть следующие. Можно выписать рекурренту в виде суммы и найти, чему она равна. Можно подставить рекурренту саму в себя и посмотреть, что получается. Можно обратиться к литературе (учебник Кормена, учебник Дасгупты)

1. $T(n) = 25T(\frac{n}{5}) + n^2$
2. $T(n) = 16T(\frac{n}{2}) + n^3$
3. $T(n) = 9T(\frac{n}{3}) + n^3$
4. $T(n) = T(n-1) + 3n$
5. $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + n$
6. $T(n) = T(\frac{n}{2}) + T(\frac{2n}{3}) + n^2$
7. $T(n) = T(n-1) + n^2$
8. $T(n) = 4T(\frac{n}{16}) + \sqrt{n}$
9. $T(n) = 9T(\frac{n}{3}) + n$

7 Выведите первый случай мастер-теоремы (основной теоремы о рекуррентных соотношениях) для целых степеней (пренебрегая округлениями). Формально, нужно показать, что если

- $T(n) = aT(\frac{n}{b}) + f(n)$, $1 \le a \in \mathbb{N}$, $1 < b \in \mathbb{R}$
- $T(n) = \Theta(1)$ для $n < N_f$
- $\exists \varepsilon > 0 : f(n) = O(n^{\log_b a - \varepsilon})$

то $T(n) = \Theta(n^{\log_b a})$.

Доказательство есть в учебниках, но задача состоит именно в том, чтобы вывести его самостоятельно. Если чувствуете в себе силы, докажите остальные случаи тоже, это полезное упражнение.

8 Дан набор из n монет, одна из которых фальшивая и легче других, и чашечные весы, на которые можно положить две группы монет одинакового размера и узнать, какая из них тяжелее, или что они равны по весу. Придумайте алгоритм, докажите его корректность и оцените асимптотику.

9 Найдите нижнюю оценку для задачи поиска фальшовой монеты, не предполагая ничего про алгоритм. **Нужно найти нижнюю оценку сложности именно для задачи, а не для какого-либо конкретного алгоритма, решающего ее.** Приведите асимптотически оптимальный алгоритм поиска фальшивой монеты. Если алгоритм из прошлой задачи уже асимптотически оптимален, ничего дополнительного делать не нужно.