

Сортировки и порядковые статистики, снова решаем вместе.

1. Рассмотрим следующий алгоритм:

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (x[i] > x[j])
            swap(x[i], x[j]);
```

Верно ли, что в результате его работы значения x всегда будут упорядочены по неубыванию или по невозрастанию? Ответ обоснуйте.

2. Рассмотрим набор слов s_1, s_2, \dots, s_n над бинарным алфавитом. Суммарная длина слов — L . Считая, что операция поменять два слова местами выполняется за $O(1)$ (не зависит от длины), предложите алгоритм их лексикографической сортировки за время $O(L)$, использующий $O(\max |s_i|)$ (максимальная длина слова) дополнительной памяти.
3. Даны n точек, равномерно распределенных в единичном круге (с центром в начале координат). Предложите алгоритм, который за $O(n)$ в среднем сортирует их по удаленности от начала координат.
4. Докажите, что сортировка элементов, использующая только сравнения с тремя исходами (меньше, равно или больше) и некоторую внутреннюю рандомизацию, имеет ожидаемое время работы $\Omega(n \log n)$. Считайте, что последовательность бит, используемая для получения случайных чисел и определяющая поведение программы определяется заранее. Таким образом, в каждый момент времени текущее состояние программы зависит от этой заранее выбранной последовательности и результатов предыдущих сравнений. Что ломается в доказательстве, если считать, что последовательность бит может быть проинициализирована от значений элементов?
5. Перенесём во внешнюю память алгоритм сортировки подсчётом. Предложите различные варианты для следующих ситуаций. Поскольку речь идёт именно про сортировку подсчётом, то считайте, что мы хотим прочитать массив не более двух раз.
- (a) Значение U (диапазон значений) меньше M (размер оперативной памяти) и элементы не содержат дополнительной информации, то есть требуется просто отсортировать массив чисел.
 - (b) Элементы содержат дополнительную информацию, которая делает их все различными, но ключи по-прежнему являются числами от 0 до $U - 1$. При каких значениях U вы можете получить сложность $O(\frac{n}{B})$?
 - (c) Значение U меньше M , но элементы содержат дополнительную информацию. Какое время работы можно получить?
6. Перенесите в модель с внешней памятью поразрядную сортировку, получив сложность $O(\frac{n}{B} \cdot \frac{\log U}{\log \frac{M}{B}})$.