

1. (1 балл) В массиве чисел  $a_1, \dots, a_n$  за одну операцию можно поменять местами любые два соседних элемента. Можно ли отсортировать массив (то есть расположить все числа в порядке неубывания) таким образом, чтобы каждый элемент поучаствовал в чётном числе перестановок? Определите ответ за  $O(n \log n)$ .
2. (1 балл) Дан массив чисел  $a_1, \dots, a_n$ , причём  $a_i \in \{0, 1, \dots, k\}$  при всех  $i$ . Отсортируйте этот массив за  $O(n + \sqrt{k})$ .
3. (2 балла) Пусть дан массив длины  $n$  и  $m$  запросов поиска порядковой статистики. Придумайте детерминированный алгоритм, отвечающий на все запросы на время  $O(n \log m)$ .
4. (1 балл) В этой задаче использовать разрешается не более одной кучи. Разработайте структуру данных  $S$ , которая бы позволяла обрабатывать любой запрос из нижеперечисленных за  $O(\log n)$ , где  $n$  — текущий размер структуры:
  - **insert**  $x$ : вставить целое число  $x$  в  $S$ ;
  - **getMin**: сообщить минимальное число в  $S$ ;
  - **getMax**: сообщить максимальное число в  $S$ ;
  - **extractMin**: удалить минимальное число из  $S$ .
5. (2 балла) Дано два отсортированных массива:  $a_1, \dots, a_n$  и  $b_1, \dots, b_m$ . Определим  $A + B$  как множество всевозможных попарных сумм вида  $a_i + b_j$  с повторениями. Например, если  $A = \{1, 2, 3\}$ , а  $B = \{2, 3, 4\}$ , то  $A + B = \{3, 4, 5, 4, 5, 6, 5, 6, 7\}$ . По данному числу  $k$  вам надо найти  $k$ -ю порядковую статистику в  $A + B$  (заметьте, что  $A + B$  вам не дан). Требуемая асимптотика:  $O(k \log k)$ . Разрешается использовать не больше  $O(k)$  дополнительных ячеек памяти.
6. (3 балла) В алгоритме поиска медианы за линейное время весь массив бьётся на блоки по 5 элементов. А что если разбивать на блоки по  $2k + 1$  элементов (где  $k$  — константа)? Найдите время работы алгоритма в таком случае.
7. (3 балла) Разработайте структуру данных  $S$ , которая бы позволяла хранить множество целых чисел, добавлять в него элементы (по одному), удалять  $|S| / 2$  наибольших элементов из множества, а также выводить максимальный элемент. Асимптотика:  $O(1)$  амортизированно (то есть  $q$  последовательных запросов к изначально пустому  $S$  должны обрабатываться за  $O(q)$ ).