1. k-й порядковой статистикой массива называется k-й по величине элемент. Придумайте алгоритм нахождения k-й порядковой статистики, работающий за $O(\log n)$ в среднем.

2. В худшем случае quicksort может потребовать $O(n^2)$ операций. Каково будет количество рекурсивных вызовов в стеке в этом случае (а следовательно и дополнительный расход памяти)? Придумайте модификацию quicksort, которая будет делать только один рекрусивный вызов вместо двух и будет иметь меньшее количество рекурсивных вызовов в стеке в худшем случае. Каковы требования к стековой памяти у модифицированного алгоритма?

3. (median-of-3) Предлагается модифицировать выбор опорного элемента следующим образом. Выберем 3 случайных элемента массива и выберем в качестве опорного элемента средний из них.

а) Какова вероятность того, что разбиение с помощью такого опорного элемента разобьет массив размера N на части размера k и $N-k$?

б) Посчитайте вероятность того, что размер каждой из частей будет не меньше $N/3$. Сравните с вероятностью этого при равномерном случайном выборе опорного элемента.

в) Предположим, что мы выбираем элементы не случайным, а фиксированным образом: первый, последний и центральный элемент массива. Разберите, каким образом такой алгоритм будет работать на массиве, отсортированном в обратном порядке.

4. (3-way quicksort)

а) Какова сложность работы quicksort на массиве, состоящем из одинаковых элементов?

б) Придумайте partitioning-алгоритм для 3-way quicksort. Для данного массива $a[\cdot]$ и опорного элемента p нужно переставить элементы массива так, чтобы он оказался разбит на 3 части:

- $a[i] < p$ для $0 \leq i \leq r$
- $a[i] = p$ для $r+1 \leq i \leq s$
- $a[i] > p$ для $s+1 \leq i \leq n-1$