

1. Эту задачу можно решить равносильно двумя способами:

1. Пробежаться по массиву 5 раз, в каждый из которых ставить в начало массива $(k - 1)$ -ый остаток от деления. Тогда сложность алгоритма будет линейной, т.к. один пробег по массиву, например, двумя указателями, имеет сложность $O(n)$. И таких пробогов будет ровно 5 (это константа, поэтому никак не повлияет на итоговую сложность), а значит итоговая сложность будет равна $5 * O(n) = O(n)$.

2. В этом варианте можно сделать 5 partition-ов с последовательно возрастающими от 0 до 4 (остатки от деления на 5) предикатами. Тогда опять же, сложность будет $O(n)$

2. В данной задаче достаточно найти $(\lceil \frac{n}{3} \rceil)$ -ую порядковую статистику по длине отрезков. Тогда ответом на задачу будут все точки, лежащие внутри $(\lceil \frac{n}{3} \rceil)$ -ого отрезка. Почему? Благодаря вложенности отрезков, все точки $(\lceil \frac{n}{3} \rceil)$ -ого отрезка принадлежат еще $\frac{2n}{3} - 1$ отрезкам, в которые этот отрезок сам вложен. Соответственно, для всех отрезков, вложенных в найденный, точки будут принадлежать еще большему количеству отрезков.

3 В данной задаче я рассматриваю быструю сортировку с детерминированным опорным элементом. В вероятностном случае будет то же самое, просто пример под "худший случай" будет более сложным. Теперь рассмотрим по пунктам:

1. В наихудшем случае нам на вход будет подан уже отсортированный по возрастанию массив. В таком случае функция рекурсивно запустится сначала от l до $r - 1$ потом до $r - 2$ и т.д. Это значит, что глубина рекурсии будет сравнима с n

2. Изменить алгоритм можно след.образом: в качестве опорного эл-та будем использовать $\frac{n}{2}$ -ую порядковую статистику, размещенную на своей позиции. В таком случае слева от нее будут эл-ты \leq ее самой, а справа - $>$. В такой ситуации глубина рекурсии будет сравнима с $\log_2(n)$

4 Для минимизации нужно найти $n + 1$ -ую порядковую статистику. Это следует из того, что минимум будет находиться в одной из точек ломаной. А именно в такой порядковой статистике потому что слева f -я будет убывать, а справа возрастать. Так случится благодаря тому, что модули до этого числа раскроются с $-$, а после него - с $+$. Тогда, все s сократятся

5 Отсортируем массив A , используя сортировку со сложностью $O(n * \log_2(n))$, например, MergeSort. Затем ставим два указателя в начало отсортированного массива и в конец. Далее идем по массиву: если сумма элементов, на которые в данный момент указывают указатели, больше нужной, то двигаем правый указатель влево на 1, иначе - вправо на 1. Идем до тех пор, пока указатели не встретятся или не будет получена необходимая сумма. Сложность: $O(n * \log_2(n)) + O(n) = O(n * \log_2(n))$. Корректность: идя по массиву таким образом, мы рассматриваем всевозможные пары слагаемых, уменьшая или увеличивая нужную сумму. Благодаря тому, что массив отсортирован, все работает.