1 Задача №1

Пусть $i = 1$, тогда, пока $a[i] < x$, будем умножать i на 2. Если $a[i] \geq x$ нужный нам элемент массива, равный x, находится слева от i, но правее $\frac{i}{2}$, так как массив отсортирован, тогда бинпоиском с границами $l = \frac{i}{2}$, $r = i$ найдем нужный нам индекс p. Алгоритм работает за $\log p + \log \frac{p}{2} = O(\log p)$.

2 Задача №2

Создадим три указателя $l_0 = 0$, $l_1 = 0$ и $r = 0$, два счетчика $cnt_0 = 0$ и $cnt_1 = 0$, а так же два массива num_0, num_1 и проинициализируем их нулями. Будем идти по массиву a, пока количество различных встреченных элементов меньше k. Будем вести учет так: на каждой итерации $num_0[a[r] - 1] + +$ и $num_1[a[r] - 1] + +$, так же, если мы добавили этот элемент впервые делаем соответственно $cnt_0 + +$ и $cnt_1 + +$, аналогично при движении указателей l_0 и l_1 убавляем соответсвующие значения. Теперь будем идти циклом от r до n и для каждого соответсвующего r будем двигать l_0 до тех пор, пока $cnt_0 > k$, и l_1 до тех пор, $cnt_1 \geq k$, двигая эти элементы обновляем соответствующие значения num и cnt. Тогда l_0 будет индексом первого элемента массива, такого что в отрезке $[l_0, r]$ ровно k различных элементов, а l_1 — первым индексом в массиве, что в отрезке $[l_1, r]$ содержится ровно $k - 1$ различный элемент. l_0 и l_1 будут посчитаны корректно на каждой итерации, так как после увеличения r количество различных элементов в отрезках могло только увеличиться. Тогда будем для каждого r прибавлять к ответу $l_1 - l_0$ — количество отрезков, в которых k различных элементов, а правая граница равна r.

Мы посчитаем все нужные нам отрезки, так как для каждого r мы считаем все подходящие отрезки, а r пробегает значения от

0 до $n-1$.

3 Задача №3

Заметим, что если $b[j-1] < a[i] < b[j]$, то $a[i] - i + j + 1$ порядковая статистика, так как ввиду упорядоченности массивов: в b — j элементов меньше $a[i]$, а в a — i элементов меньше $a[i]$. Тогда будем выбирать такие i и j, что $i + j + 1 = k$.
Если $b[j-1] < a[i] < b[j]$ или $a[i-1] < b[j] < a[i]$, то $a[i]$ или $b[j]$ соответственно и есть k-я порядковая статистика. Пусть для определенности массив a меньше и $n < m$, если иначе, просто поменяем массивы местами.
Заведем 2 переменные $l = 0$, $r = n - 1$, и будем бинпоиском искать нужный нам идекс $i = \frac{l+r}{2}$, и соответсвенно j, если выполнится одно из двух условий написанных выше — выйдем из бинпоиска и выведем данный элемент, так как он и есть ответ. Докажем корректность бинпоиска: если оба условия выхода из бинпоиска не выполнены и:

1. $a[i] < b[j-1] < b[j]$, тогда в позициях от l до i может находиться максимум $i + j$ - я порядковая статистика, тогда продолжать бинпоиск в подмассиве $a[l, i]$ бессмыслено, так как $i + j < k$ по условию, тогда продолжаем бинпоиск в подмассиве $a[i + 1, r]$.

2. $a[i] > b[j] > b[j-1]$, то в позициях от i до r может находится минимум $i+j+2$ - я порядковая статистика, тогда продолжать бинпоиск в правой части массива $a[i, r]$ бессмыслено, так как $i+j+2 > k$ по условию. Тогда продолжаем бинпоиск в подмассиве $a[l, i-1]$.

Алгоритм отработает за $O(\log\min(n, m))$ так как мы делаем бинпоиск в меньшем массиве.