1. Найдем $k : 2^k < p, 2^{k+1} >= p$
   Его можно легко найти за $\mathcal{O}(\log_2 p)$ просто итеративно увеличивая k и проверяя условие.
   Теперь у нас есть отрезок отсортированного массива длины $\mathcal{O}(p)$ на котором нужно найти элемент. Сделать это можно бинпоиском за $\mathcal{O}(log_2 p)$

2. Это противная задача. Давайте научимся решать другую.
   А именно, найдем количество пар (L, R), таких что на этом отрезке не более k различных чисел.
   Пусть мы зафиксировали левую границу L, тогда пусть $R(L)$ — минимальная граница, такая что на $[L; R(L)]$ больше чем k различных чисел.
   Получается, для фиксированной границы L, ответом будет $R(L) - L$
   Заметим что $\forall L < n \; R(L) \leq R(L+1)$
   Давайте запомним $R(L)$ и будем считать $R(L+1)$ используя предыдущее.
   Нужно ещё как-то поддерживать количество различных элементов на отрезке, а так же уметь увеличивать правую и левую границы.
   Заведём массив подсчета cnt. Будем в нем хранить количество элементов по значению.
   Тогда если мы добавляем a_i и $cnt_{a_i} = 0 \implies$ у нас добавился новый уникальный элемент. Аналогично при удалении: если удаляем элемент, а он один в отрезке, то значит кол-во различных уменьшилось на 1.
   Каждый раз мы двигаем правую границу отрезка, но суммарно более n раз мы подвинуть не можем, следовательно амортизированное время работы $\mathcal{O}(n)$.
   Посчитали количество пар (L, R) таких что количество различных чисел на этом отрезке не более k.
   Посчитаем тоже самое для $k - 1$.
   Вуаля, если вычтем из первого второе, то получим количество пар (L, R) таких что на этом отрезке ровно k различных чисел.

3. Вот у нас есть два массива a, b размерами n, m и мы ищем k-ю порядковую статистику. Далее нумерация будет с 1
   Пусть $i = \frac{n}{2}, j = k - \frac{n}{2}$
   Заметим что $i + j = k$
   Посмотрим на a_i, b_j
   Если $b_j < a_i < b_{j+1}$, то ответ a_i, потому что он стоит ровно на месте $i + j = k$
   Если $a_i < b_j < a_{i+1}$, то ответ b_j (аналогично)
   Теперь, если оба условия не выполнились и $a_i > b_j$, то $a_i > b_{j+1}$, так как иначе было бы верно второе условие. Отсюда видно что никакой из элементов до b_j включительно (в массиве b конечно) не может быть k-й порядковой статистикой, так как все они будут с индексами меньшими $i+j$. Можно заметить что и все элементы большие a_i (в массиве a) тоже не будут k-й порядковой статистикой, потому что будут иметь индексы большие $i + j$ в массиве слияния.

Тогда мы можем скипнуть j элементов массива b и искать $k-j$ порядковую статистику на массивах $a[1 : i-1], b[j+1 : m]$
Симметричный случай, когда $b_j > a_i$, разбирается аналогично.
Что делать, когда один из массивов длины 1?
Пусть $n = 1$, тогда если $b_{k-1} < a_0 \Rightarrow$ ответ — a_0, иначе ответ — b_{k-1}
Если $m = 1$, то поменяем массивы местами и всё аналогично.
Каждый уровень рекурсии n уменьшается в 2 раза, значит если передать первым массивом — меньший из двух, то алгоритм будет работать за $\mathcal{O}(\log_2(\min(n, m)))$