Homework

Panov Ivan, M3139

11-09-2019

Задача 1

Данную задачу можно решать алгоритмом бинарного поиска, но если запустить его на всем массиве, то получим асимптотику $O(\log(n))$. Чтобы уменьшить время работы понадобится сузить область, на которой запускается бинарный поиск. Давайте перебирать степени двойки начиная с нулевой и проверять, что элемент на позиции $a[2^i] \le x$ (i - перебираемая степень), когда условие перестанет выполнятся или $a[2^i]$ станет больше n, остановим подбор степени. Таким подбором мы найдем отрезок в котором лежит x за $O(\log(p))$, так как увеличивали каждый раз отрезок в 2 раза и остановились после, того как x стал входить в отрезок. Теперь воспользуемся бинарным поиском: будем каждую итерацию алгоритма делить текущий отрезок на две равные части и проверять, в какой лежит x. Так как отрезок имеет длину не большую чем $2p$, бинарный поиск сделает $O(\log(p))$ итераций. Суммарная асимптотика: $O(\log(p))$.

Задача 2

Заведем два массива на n элементов $cnt1[\ ]$ и $cnt2[\ ]$, изначально заполненные нулями, в i-ой ячейке которых будем хранить сколько чисел со значением i рассматривается на данный момент. Дополнительно заведем переменные $num1$ и $num2$ равные изначально n, в которой будем хранить количество не нулей в $cnt1$ и $cnt2$, заметим, что num-ы по факту хранят количество различных рассматриваемых элементов.
Для решения данной задачи будем поддерживать три указателя: $p1$ - указатель на начало отрезка, $p2$ - указатель на конец последнего слева неподходящего отрезка, $p3$ - указатель на конец последнего слева подходящего отрезка. Пусть ответ для какого-то префикса посчитан, рассмотрим переход к следующему отрезку. Требуется сдвинуть левую границу на 1 вправо. При сдвиге изменится значения в $cnt1$ и $cnt2$, их надо пересчитать, вычтем из $cnt1[a[p1]]$ единицу, и если $cnt1[a[p1]]$ стало нулем, вычтем единицу из $num1$, проведем такие же операции с $cnt2[\ ]$ и $num2$, увеличим $p1$. Теперь надо правильно подобрать правые границы. Будем двигать $p2$ пока $num1 < k$: прибивим в $cnt1[a[p2]]$ единичку, и если до увеличения $cnt1[a[p2]]$ было нулем, прибавим один в $num1$. Будем аналогично двигать $p3$, пока $num2 \le k$, только теперь будем обновлять $cnt2[\ ]$ и $num2$. Теперь можно обновить ответ: $ans+ = (p2 - p1)$.
Данный алгоритм работает за $O(n)$, так как мы рассмотрим каждый элемент не более четырех раз: при добвалении в отрезоки и при удалении.

Задача 3

n - Длина массива с большей длиной, m - с меньшей. Давайте искать бинарным поиском количество элементов, которых возьмем в массиве с меньшей длиной. Рассмотрим одну итерацию бинарного поиска. Пусть мы определили, что количество элементов, которых надо взять в меньшем массиве, лежит в отрезке от l по r. Определим $mid = \frac{l+r}{2}$. Допустим мы взяли mid элементов в меньшем массиве и $k - mid$ элементов во втором. Рассмотрим последнии взятые в обоих массива: $p1$ - в меньшем, $p2$ - в большем. Если $p1 \leq p2$ сдвинем l: $l = mid$. Этот переход коректен, так как все элементы в меньшем массиве, лежащие на префиксе до l, меньше, чем $p2$, то взяв меньший префикс, мы не учтем некоторые элементы, которые меньше нынешнего максимального, следовательно k-ая порядковая статистика будет посчитана некоректно. Если $p1 > p2$ сдвинем r: $r = mid$(аналогично рассуждения выше). Начальное зачение $l = \max(k-n, 0)$, $r = \min(n, m)$. После того как найдем количество элементов в меньшем, которое нужно взять, выберем ответ. Ответом будет являться максимум из $p1$ и $p2$, так как мы выбрали в ровно k самых маленьких из обоих массивов.