Домашняя работа №2

Шибаев Александр Б05-222

Октябрь 2022

1 Первая задача

Мы умеем находить количество элементов слева, больших данного (такая задача была в контесте), значит и умеем находить количество элементов справа, меньших данного. Для этого мы используем MergeSort. При этом, если $a = x_i$ стоял левее $b = x_j$ и $a > b$, то мы поменяем их между собой местами нечетное число раз, т.к. в отсортированном массиве a стоит правее b отсортированном массиве. Аналогично если $a = x_i$ стоял правее $b = x_j$ и $a < b$. Если $a = x_i$ стоял левее $b = x_j$ и $a < b$, то и в отсортированном массиве a стоит левее b, поэтмоу a и b мы будем менять четное число раз. Поэтому если у x_i $\forall i$ количесво количество элементов слева, больших x_i + количесво количество элементов справа, меньших x_i - четное, то ответ - да, иначе ответ - нет.

2 Вторая задача

Пусть $k = \overline{k_1 k_2 k_3 k_4 ....k_n}$.
Тогда если $a_i = \overline{a_{i_1} a_{i_2} a_{i_3} ... a_{i_n}}$ (Первые a_{i_j} могут быть нулями), то мы разбиваем это число на два блока - $x_1 = \overline{a_{i_1} a_{i_2} a_{i_3} ... a_{i_{n/2}}}$ и $x_2 = \overline{a_{i_{n/2+1}} a_{i_{n/2+2}} a_{i_{n/2+3}} ... a_{i_n}}$. И теперь сортируем числа "поразрядной"сортировкой.
Т.к. "половинки"чисел a_i не больше, чем $\sqrt{a_i} <= \sqrt{k}$, то ассимптотика такой сортировки $O(n + \sqrt{k})$.

3 Третья задача

Пусть $a[n]$ - массив чисел, а $q[m]$ - массив запросов.
Если $m \geq n$, то просто отсортируем массив чисел за $O(n \log n)$. И $\forall i \leq m$ будем выводить $a[m[i]]$.
Если $m < n$, то отсортируем массив q за $O(n \log n)$(запомним их первоначальный порядок). Теперь возьмем серединный элемент: $p = q[(r + l)/2]$ (изначально $l = 0, r = m$). Найдем p-тую порядковую статистику за $O(n)$. Теперь все числа стоящие левее $a[p]$ - меньше или равны $a[p]$, а стоящие правее - больше или равны. Поэтому теперь мы запускаемся рекурсивно от левой и правой половины массива q, но для левой половины нам нужно рассматривать только отрезок массива a - $[l_a, p]$, а для правой половины - $[p + 1, r_a]$. Т.к. q отсортирован, то все порядковые статистики из левой половины q будут в отрезке $[l_a, p]$ массива a.(изначально $l_a = 0, r_a = n$). Глубина такого алгоритма - $\log m$, и на каждом слое мы проходим по нескольким непересекающимся отрезкам массива a, причем их суммарная длина - n. Поэтому ассимптотика такого алгоритма - $O(n \log m)$.

4 Четвертая задача

Будем хранить обычную $min-$кучу и поддерживать максимум в ней. Тогда при удалении минимума, максимум меняться не будет(т.к. у нас $min-$куча). Если куча становится пустая, то и максимум пустой. При добавлении нового элемента мы просто сравниваем его с максимумом и если нужно, то изменяем максимум, и кладет его в кучу стандартным способом. Удаление минимума из $min-$кучи - $O(logn)$, добавление тоже за $O(logn)$, получение максимума и минимума за $O(1)$

5 Пятая задача

В этой задаче будем использовать опять min-heap. В каждой вершине будем хранить тройку чисел - $(i,\ j,\ a_i + b_j)$. Тогда если минимум - $(i,\ j,\ a_i + b_j)$, то будет класть в кучу элементы $(i + 1,\ j,\ a_{i+1} + b_j)$ и $(i,\ j + 1,\ a_i + b_{j+1})$, и удалим минимальный элемент - $(i,\ j,\ a_i + b_j)$ и положим его в ans - массив ответов. Т.к. массивы A и B отсортированы, то мы добавляем числа, не меньшие, чем минимальный элемент. После этих операций, количесвто элементов в куче увеличиться на 1, и количество элементов в ans учеличиться на 1. И при этом в ans всегда на один элемент меньше, чем в куче. Поэтому когда в куче оказалось k элементов и минимальный элемент там - $(i,\ j,\ a_i{+}b_j)$, то мы просто кладем его в ans, таким образом в ans ровно k наименьших элементов $\Rightarrow$ мы нашли k порядковую статистику. Каждый раз в куче не более k элементов, и мы делаем $2k$ добавлений $\Rightarrow$ ассимптотика такого решения $O(k \log k)$.