

## Задача А. Преступление без наказания

Пусть  $S$  — сумма всех  $a_i$ -х. Тогда сухих листов бумаги осталось ровно  $k - S$ .

## Задача В. Много половин

Если  $k$  сыновей можно собрать в одной точке, то это значит, что полуплоскости этих принцев пересекаются хотя бы в одной точке. Таким образом, задача сводится к поиску максимального количества полуплоскостей, пересекающихся хотя бы в одной точке.

Разделим все полуплоскости на 2 группы: в первой будут типы 1 и 2, во второй - 3 и 4. Заметим, что если полуплоскости из первой группы имеют хотя бы одну точку пересечения, то они пересекаются хотя бы по прямой  $x = b$  для некоторого  $b$ . Аналогично для полуплоскостей из второй группы: если они имеют хотя бы одну точку пересечения, то они пересекаются хотя бы по прямой  $y = c$  для некоторого  $c$ . Тогда если полуплоскости из первой группы имеют хотя бы одну общую точку и полуплоскости из второй группы имеют хотя бы одну общую точку, то точка  $(b, c)$  принадлежит пересечению всех полуплоскостей из двух групп.

Решим задачу независимо для каждой из групп, после чего сложим ответы. Решение для первой группы (для второй аналогично):

1. В массив  $left$  сложим все числа  $a$  полуплоскостей первого типа, в массив  $right$  —  $a$  полуплоскостей второго типа.
2. Отсортируем массивы  $left$  и  $right$ .
3. Будет идти по массиву  $right$  от наименьшего  $a$  к наибольшему. Тогда если в ответе есть все полуплоскости из  $right$  от 0-й до  $i$ -й включительно, то их пересечение —  $i$ -я полуплоскость. Аналогично для  $left$ : пересечение всех полуплоскостей от  $j$ -й до конца —  $j$ -я полуплоскость.  $i$ -я полуплоскость из  $right$  и  $j$ -я полуплоскость из  $left$  пересекаются, если  $right[i] \leq left[j]$ . Значит можно для каждого  $i$  найти минимальное  $j$  такое, что  $left[j] \geq right[i]$ . Для этого будем перебирать все  $i$  от 0 до  $|right| - 1$  и поддерживать  $j$  такое, что  $left[j] \geq right[i]$ . При увеличении  $i$ ,  $j$  будет только увеличиваться, поэтому можно использовать два указателя на массивах  $left$  и  $right$  —  $O(n)$ . Для пары  $(i, j)$  ответ —  $(i + 1 + |left| - j)$ .

## Задача С. Очаровательные числа

Чтобы отвечать на запрос на подотрезке  $[l, r]$ , научимся для начала отвечать на запрос на префиксе  $[1, x]$ . Тогда кол-во подходящих чисел на  $[l, r]$  будет равно кол-ву таких чисел на отрезке  $[1, r]$  минус кол-во таких чисел на отрезке  $[1, l - 1]$ .

Дальше, пронумеруем цифры в числе  $n$  длины  $m$  в порядке слева направо. Напомню, что  $f(n) = (10a_1 + a_2) \cdot \dots \cdot (10a_{m-1} + a_m)$ .

Если в  $i$ -ом разряде стоит 0, так как  $i \neq 1$ , в произведении присутствует множитель  $10a_{i-1} + a_i = 10 \cdot a_{i-1}$ . Но тогда, если всего в числе было  $k$  нулей, то  $f(n) \vdots 10^k$ , и из  $f(n) = n \implies n \vdots 10^k$ , а все эти  $k$  нулей стоят у  $n$  на конце (в последних  $k$  разрядах).

Далее, если в  $n$  есть цифра «00», то произведение будет 0, и  $f(n) \neq n$ . Отсюда  $k \leq 1$ , то есть, все множители  $10a_i + a_{i+1} \geq 10 \cdot 1 + 0 = 10$ .

Осталось заметить, что при  $y \geq 1$  и  $x \neq 0$  выполнено  $\overline{xy} \cdot \overline{yc_1 \dots c_t} = x \cdot \overline{yc_1 \dots c_t}0 + y \cdot \overline{yc_1 \dots c_t} \geq x \cdot \overline{10 \dots 00} + 1 \cdot \overline{yc_1 \dots c_t} = \overline{xy c_1 \dots c_t}$ . И равенство достигается только при  $y = 1, c_1 = \dots = c_t = 0$ .

Теперь докажем, что при  $k \leq 1$  у нас  $f(n) \geq n$ . По индукции, базу для  $n = 3$  доказал в предыдущем абзаце. А для  $n \geq 4$  получаем  $f(n) = f(\overline{a_1 \dots a_n}) = (10a_1 + a_2) \cdot f(\overline{a_2 \dots a_n}) \geq (10a_1 + a_2) \cdot \overline{a_2 \dots a_n}$ , но  $a_3 > 0$ , и получаем строгое неравенство.

Отсюда очаровательными могут быть только числа из  $n \leq 3$  знаков (то есть, числа не больше 1000).

Таким образом, одним из вариантов решения может быть, например, перебрать сначала все числа до 1000, проверив каждое на очаровательность, и сохранить подходящие в отдельный массив,

а потом для ответа на очередной запрос  $[l, r]$  пробежать все числа из этого массива, считая кол-во из них, лежащих внутри  $[l, r]$ , и выводить получившееся число.

## Задача D. Полузабытое ПСП

Добавим внешние скобки  $()$  к нашей посл-ти:  $s \rightarrow (s)$  (очевидно, ответ не изменится). Скажем, что всей этой строке соответствует какая-то корневая вершина  $0$ .

Теперь по рекурсивному определению ПСП можно построить дерево следующим образом: если текущая подстрока  $A$  соответствует вершине  $i$  и имеет вид " $A_1 A_2 \dots A_m$ " (за " $"$ " обозначим какую-то пару скобочек из  $()$ ,  $[]$  или  $\{\}$ ), проведём ребра из  $i$  в вершины строк  $A_1, A_2, \dots, A_m$  (которые также имеют вид "ПСП"). В частности, при  $m = 0$  (если  $A = ""$ ) закончим вызывать рекурсию.

Тогда каждой вершине соответствует какое-то кол-во  $|$ , находящихся «на этом» уровне рекурсии, то есть  $|$ , которые находятся непосредственно внутри текущей пары скобок (и не находящихся внутри никакой пары скобок в поддереве).

А теперь ключевое соображение - после замены  $|$  на  $/$  и  $\backslash$  посл-ть будет ПСП тогда и только тогда, в каждой вершине независимо мы, в итоге, получим ПСП из заменённых  $|$  (если итоговая посл-ть без  $|$  была ПСП).

Свели к тому, что есть набор чисел с суммой до  $10^6$ , и нам нужно перемножить кол-ва способов для каждого  $i$  сделать ПСП из  $a_i$  скобок.

Это уже стандартная задача, которая при маленьких ограничениях решается dp-кой, а вообще является последовательностью чисел Каталана (для  $n : 2 \rightarrow \frac{C_n^{0.5n}}{0.5n+1}$  способов).

Хинт в реализации: вместо построения дерева будем идти со стеком, добавляя в него открывающиеся скобки и удаляя последнюю, если текущая скобочка закрывающаяся и парная ей. Тогда, если все удаления корректны (стек не пуст и парные), а в конце он пуст - ПСП.

И добавляя все  $|$  к последним на данный момент открывающимся скобочкам в стеке (после преобразования  $s \rightarrow (s)$  они всегда есть), мы получим в конце нужный нам массив длин ПСП-шек.

## Задача E. Чудесные пары

Давайте построим корневую декомпозицию. Разобьём массив на блоки длины  $\sqrt{n}$ , тогда их количество будет равно  $\frac{n}{\sqrt{n}} = \sqrt{n}$ .

Применим сжатие координат — теперь наши числа не превосходят  $n$ .

Для каждого блока будем хранить  $pref_{i,j}$  - количество числа  $i$  на первых  $j$  блоках (другими словами, сколько раз число  $i$  встречается в первых  $j$  блоках). Это займёт  $\max(a_i) * \sqrt{n}$  памяти.

Заметим теперь, что мы можем узнать количество конкретного числа на подотрезке из полных блоков за  $O(1)$ .

Далее предпросчитаем количество пар равных для блоков с  $i$  по  $j$ , пусть это будет  $cnt_{i,j}$  (работает за  $n * \sqrt{n}$ ).

Теперь для ответа на запрос нам достаточно взять блоки, которые полностью входят в отрезок  $[l, r]$ , а также, возможно, некоторые неполные блоки. Последних будет не больше двух, и их длина не превосходит  $\sqrt{n}$ . Давайте просто пройдемся по этим числам и для каждого прибавим количество пар, которые ещё не были учтены с ним. Для этого можно, например, завести глобальный массив  $used_x$  - количество элемента  $x$  на оставшихся двух неполных блоках. Насчитать  $used_x$  можно за  $O(\sqrt{n})$ , ввиду того, что максимальный размер неполного блока равен  $\sqrt{n}$ .

Тогда если обозначить за  $S$  количество числа  $x$  на полных блоках, которые уже учтены (за  $O(1)$  с помощью  $pref$ ), то к ответу достаточно прибавить:  $(S + used_x) * (S + used_x + 1) / 2 - S * (S + 1) / 2$ .

Осталось пробежаться за  $\sqrt{n}$  и сделать  $used_x = 0$ .

Итого  $O((n + q) * \sqrt{n})$  в онлайн.