0.1 B. Hats

Будем идти по $i \in [0; n)$ и хранить отрезок таких k, что мы могли использовать ровно k шляп до этого момента. (заметим, что такие k действительно образуют отрезок)

Изначально доступно только $k = 0$.

Если $s_i = 1$, то нужно посмотреть на позицию p числа i, если мы могли использовать в точности отрезок до p, то теперь мы можем использовать только ровно все шляпы до этой включительно, а иначе ответ 0.

Если $s_i = 0$, то нужно посмотреть на максимальное количество шляп, которые мы могли использовать до этого момента, и если номер следующей шляпы не совпадает с i, то увеличить это значение на 1.

0.2 C. Pepeland

Заметим, что ответ всегда существует.

Выберем какое-нибудь остовное дерево, и подвесим его за какую-нибудь вершину.

Покрасим корень в цвет, которого нет ни у какого ребра.

Дальше будем красить вершины в порядке обхода dfs.

Пусть у вершины u родитель в дереве p, и ребро между ними имеет цвет c. Если цвет p это c, то покрасим u в любой другой цвет, а иначе покрасим u в цвет c.

0.3 D. Removals

Пусть dp_{ij} – ответ на задачу для префикса длины i, если мы удалили ровно j чисел.

$\forall 0 \le i < n$, из dp_{ij} мы всегда можем перейти в dp_{i+1j} за a_i.

Пусть p – номер ближайшей слева к i позиции, которая есть в b, если такая есть, и $-inf$, если такой нет.

Удалить a_i мы можем тогда и только тогда, когда $j < k$ и $i - j \le p$.

Если мы можем удалить a_i, то есть переход из dp_{ij} в dp_{i+1j+1} за 0.

Ответ это максимум dp_{nj} по $0 \le j \le k$.

Получаем решение за $O(nm)$

0.4 E. Databases

Будем делать сканлайн по $i \in [0; n)$ и поддерживать те запросы, которые были применены к текущей позиции i.

А именно, развернём номера запросов, и будем поддерживать дерево Фенвика на $[0; q)$ с единицами для тех запросов, которые сейчас активны и нулями для остальных.

Ещё будем для каждого значения хранить *set* активных запросов, которые добавляли это значение, и ещё одно дерево Фенвика на $[0; q)$ с единицами для тех активных запросов, которые являются первыми (в развернутом порядке) для своего значения.

Тогда чтобы найти количество различных элементов в i-й очереди в конце, нам нужно найти в первом дереве Фенвика минимальный префикс с суммой $\geq c_i$, и посчитать сумму на этом префиксе во втором дереве Фенвика.

В итоге получаем решение за $O((n + q)log(n + q))$.

P.S.

Опишем отдельно, как искать в дереве Фенвика минимальный префикс с суммой $\geq w$ (если такого нет, то мы будем возвращать q).

Здесь мы пользуемся реализацией дерева Фенвика с 0-индексацией, где вершина i отвечает за отрезок $[i\&(i + 1), i]$

Будем восстанавливать ответ по битам от старших к младшим.

Пусть мы нашли все биты ответа в разрядах старше b, и они образуют число u.

Пусть f – дерево Фенвика, в котором мы искали минимальный префикс с достаточно большой суммой.

Пусть w – сумма, которую нам ещё осталось набрать (то есть разность изначального значения w и суммы на префиксе $[0; u)$)

Если $u + 2^b - 1 < q$ и $f[u + 2^b - 1] < w$, то ответ $\geq u + 2^b$, то есть в разряде b точно стоит 1. Тогда добавим этот бит в ответ и уменьшим w на $f[u + 2^b - 1]$ (это сумма на отрезке $[u; u + 2^b)$)

Иначе, ответ $< u + 2^b$, то есть в разряде b точно стоит 0.

Сделав так для всех b, получаем ответ.