Бизнес-шоу

$q = 1$ (10 баллов)

Мы не можем не активировать единственное возможное предложение. Теперь известно, какие клетки включены, а какие нет. Так что мы можем посчитать ответ с помощью стандартной динамики (путь максимальной стоимости в прямоугольнике).

Решение за $O(n^2q)$ (16 баллов)

Пусть $pref[i][j]$ — сумма первых j элементов в iм ряду. Тогда определим $s[i] := pref[0][i+1] - pref[1][i]$, $f[i] := pref[1][i+1] - pref[2][i] + pref[2][n]$.

Заметим, что если мы вошли во второй ряд в клетке $(2, i)$ и вышли из него в клетке $(2, j)$, то суммарное изменение баланса от прохождения по клеткам равно $s[i] + f[j]$.

Теперь заметим, что теперь задача свелась к тому, чтобы максимизировать $s[i] + f[j] - cost(i, j), i \leqslant j$, где $cost(i, j)$ — минимальная стоимость разблокировки отрезка $[i, j]$.

Теперь посчитаем $dp[i]$ — максимальная выгода, которую можно получить, дойдя до клетки $(2, i)$, учитывая стоимости отрезков, которые мы разблокировали.

Пересчет: переберем отрезок, покрывающий i, и прорелаксируем $dp[i]$ значением $\max\limits_{l-1 \leqslant j < i} dp[j] - k$. Также мы можем войти во второй ряд в точке i, тогда прорелаксируем через $\max\limits_{l \leqslant j < i} s[j] - k$.

Ответом будет $\max\limits_{0 \leqslant i < n} f[i] + dp[i]$

Решение за $O(n\,q\,log\,n)$ (+14 баллов)

Заметим, что для пересчета динамики из предыдущего решения можно воспользоваться деревом отрезков, и тогда вся динамика будет пересчитываться за $O(n\,q\,log\,n)$

Все k_i одинаковы (+17 баллов)

Будем оптимизировать пересчет динамики. Заметим, что для предложения l, r, k оно будет корректно для пересчета с момента, когда пересчитываем $dp[l]$ до момента, когда пересчитываем $dp[r]$ включительно.

Теперь у нас есть запросы вида «Добавить/удалить отрезок для пересчета динамики» и «Посчитать $dp[i]$».

Заметим, что добавить отрезок, это то же самое, что сказать, что можно использовать все $dp[j]$ на отрезке для перехода стоимости k.

То есть, задача теперь свелась к запросам «Включить/выключить отрезок динамики» и посчитать минимум по включенным значениям на отрезке. Это все можно реализовать с помощью дерева отрезков за $O(log\,n)$ на запрос

Решение за $O(n\,log^2 n)$ (+21 балл)

Обобщим подход для всех одинаковых k_i.

Заметим, что чтобы пересчитать динамику, достаточно для каждого j пересчитываться только через самый дешевый отрезок. Теперь запросы имеют вид «Добавить/удалить возможность перехода из $dp[j]$ на отрезке за k рублей» и «Посчитать максимум по $dp[j]$ − (минимальная стоимость перехода из j в i)».

Чтобы отвечать на эти запросы достаточно хранить в каждой вершине дерева отрезков `set` из стоимостей предложений, в который входит соответствующий этой вершине отрезок, и при пересчете ответа для вершины брать из этого `set` минимальный элемент и максимальное dp на этом отрезке.

Полное решение.

Посчитаем $dp[i]$ — аналогичная динамика, но известно, что самый правый отрезок, который мы взяли, заканчивается в i.

Теперь переходов суммарно $O(q)$, посчитаем эту динамику за $O(q\,log\,n)$.

Рассмотрим множество отрезков, которое является оптимумом (при его включении достигается максимальная выгода).

Переберем отрезок, который может быть самым правым в этом оптимуме. Заметим, что максимальная выгода, если этот отрезок самый правый, и есть хотя бы один другой отрезок, который мы взяли, будет равен $\max\limits_{l \leqslant i \leqslant j \leqslant r} dp[i] + f[j] - k$ (предпоследний отрезок закончился в i, вышли из j). Для каждого отрезка эту величину можно посчитать с помощью дерева отрезков за $O(log\,n)$.

А если берем только один отрезок, то надо прорелаксировать ответ $\max\limits_{l \leqslant i \leqslant j \leqslant r} s[i] + f[j] - k$.

Итого имеем решение за $O(q\,log\,n)$.