Задача A. Раскраска автобусов

Сначала научимся делать следующее: по данному n научимся находить количество краски, требующейся на первые n столбиков — с 1 по n-й включительно. Обозначим это количество краски за $f(n)$. Если мы научимся его быстро находить, то ответ на задачу, очевидно, будет равен $f(r)-f(l-1)$.

Как найти $f(n)$? Во-первых, все столбики с нечетными номерами имеют высоту ноль. Удалим их все, т.е. сместим четные столбики к началу картины так, чтобы между ними не оставалось пустого места — чтобы они были нарисованы в два раза плотнее, чем были сначала. У нас осталось $\lfloor \frac{n}{2} \rfloor$ столбиков (скобки $\lfloor\rfloor$ обозначают округление вниз, т.е. взятие целой части), и у каждого такого столбика высота как минимум 1. Отрежем у каждого столбика нижний квадратик. После этого несложно видеть, что мы получим кусок той же картины столбиков, с которой мы начинали — только теперь у нас будет изображено $\lfloor \frac{n}{2} \rfloor$ столбиков. Тогда их суммарная площадь равна $f(\lfloor \frac{n}{2} \rfloor)$. Вспомним, что мы отрезали у $\lfloor \frac{n}{2} \rfloor$ столбиков нижние квадратики — поэтому окончательно получаем

$$f(n) = \lfloor \tfrac{n}{2} \rfloor + f(\lfloor \tfrac{n}{2} \rfloor)$$

Данная функция уже легко вычисляется рекурсивно; можно также написать простой цикл while. Отметим также, что при маленьких l и r задачу можно было решить, просто перебрав все столбики от l до r и вычислив их высоту; такое решение тоже набирало некоторые баллы.

Задача B. Выбор полосы

разбор появится позднее

Задача C. Разрезы

В первых двух подзадачах первые два типа запросов будем обрабатывать тривиально: для каждого индекса i будем хранить есть ли между i и $i+1$ элементами разрез. Теперь давайте рассмотрим как обрабатывать запросы третьего типа.

Для решения первой подгруппы достаточно просто перебирать концы предполагаемого отрезка, считая сумму на каждом из отрезков и проверяя не содержит ли он разрезов. Сложность такой наивной реализации будет $\mathcal{O}(q \cdot n^2)$.

Перед тем как перейти ко второй подзадаче давайте обсудим такую задачу: найти в массиве подотрезок с максимальной суммой. Эту задачу можно решать так: будем идти по массиву и накапливать в некоторой переменной s текущую частичную сумму. Если в какой-то момент s окажется отрицательной, то мы просто присвоим $s = 0$. Утверждается, что максимум из всех значений переменной s, случившихся за время работы, и будет ответом на задачу. Давайте применим этот алгоритм к нашей задаче, не забывая еще проверять каждый отрезок на корректность. Тогда мы получим решение за $\mathcal{O}(q \cdot n)$.

Для решения третьей подзадачи достаточно заметить, что ответом на третий запрос является сумма на отрезке. Соответственно, для вычисления суммы на отрезке можно воспользоваться префиксными суммами.

Для решения четвертой группы можно воспользоваться модифицированным деревом отрезков. В каждой вершине этого дерева нужно хранить четыре величины: сумма на отрезке этой вершины, максимальная сумма на префиксе этой вершины, максимальная сумма на суффиксе этой вершины и ответ на отрезке, который соответствует этой вершине. Несложно понять, что все эти величины можно пересчитать с помощью таких же величин в детях.

Для полного решения можно создать фиктивные элементы, которые будут отвечать за разрезы. Соответственно значения в них будут равны $-\infty$, если разрез существует и 0 в противном случае. Итоговая асимптотика будет равна $\mathcal{O}(q \cdot \log n)$.

Задача D. Боулинг

Переформулируем задачу на эквивалентую: посчитать количество способов выбрать допустимую последовательность шаров и одновременно выбрать k (необязательно различных) элементов с весом не более y из этой последовательности.

Воспользуемся динамическим программированием для решения новой задачи. Пусть $\mathrm{dp}[i][j][h]$ — количество последовательностей длины i, чья сумма весов по модулю w сравнима с j, и которые содержат ровно h выбранных элементов с весом y или меньше.

Для упрощения вычислений можно также считать что $\mathrm{dp}[i][j-w][h] = \mathrm{dp}[i][j][h]$. Пересчет устроен так $\mathrm{dp}[i][j][h] = \sum_{p=0}^{w-1} v_p \mathrm{dp}[i-1][j-p][h] + \sum_{p=0}^{y-1} \sum_{q=0}^{h} \binom{h}{q} v_p \mathrm{dp}[i-1][j-p][h-q]$. Наивный пересчет работает за $\mathcal{O}(Nw^2k^2)$.

Для полного решения воспользуемся техникой «разделяй и властвуй». В случае нечетного i сделаем такой же пересчет, в случае четного рассмотрим последовательности в два раза меньше, а затем объединим их следующим образом: $\mathrm{dp}[i][j][h] = \displaystyle\sum_{p+p' \equiv_w j} \sum_{q+q'=h} \binom{h}{q} \mathrm{dp}[i/2][p][q] \cdot \mathrm{dp}[i/2][p'][q']$. Итоговая асимптотика $\mathcal{O}(w^2k^2 \log n)$.