

Прямоугольное Пятно

Автор задачи: Даниил Орешиников, разработчик: Константин Бац

Подойдем к задаче с другой стороны. Сначала рассмотрим версию задачи, в которой прямоугольники нельзя поворачивать. Тогда для любых двух идущих подряд выбранных прямоугольников i и j должно выполняться, что $w_i \leq w_j$ и $h_i \leq h_j$. Сфокусируемся на первом условии и сделаем так, чтобы оно выполнялось автоматически: отсортируем все прямоугольники по возрастанию h_i (а при равенстве — по возрастанию w_i).

Теперь задача свелась к тому, чтобы выбрать такую подпоследовательность прямоугольников, в которой для любых двух подряд идущих i и j выполняется $h_i \leq h_j$. А это уже стандартная задача о наибольшей возрастающей последовательности (НВП), которая решается за время $\mathcal{O}(n \log n)$. Остается решить проблему того, что мы не учли возможность поворачивать прямоугольники на 90° .

Заметим, что все прямоугольники, не являющиеся квадратами, можно скопировать, поменяв местами ширину и высоту, и это не повлияет на ответ задачи. Действительно, поворачивать квадраты нет смысла — они не изменятся, а если у нас есть и прямоугольник (h_i, w_i) , и прямоугольник (w_i, h_i) , то максимум один из них войдет в ответ — при $h_i \neq w_i$ ни один из них нельзя вложить в другой.

Тогда выполним описанное выше действие и отсортируем прямоугольники, например, по ширине. Тогда задача сводится к поиску наибольшей возрастающей последовательности высот прямоугольников, как мы уже отметили ранее.

Один из стандартных способов нахождения НВП за время $\mathcal{O}(n \log n)$ с восстановлением ответа — это динамическое программирование с двоичным поиском. Будем строить следующую динамику: для $0 \leq i \leq n$ пусть $d[i]$ — наименьшее число, на которое может оканчиваться возрастающая последовательность длины i .

Изначально мы предполагаем, что $d[0] = -\infty$, а все остальные элементы $d[i] = +\infty$ — после последовательности длины 0 можно поставить что угодно, а вот последовательности положительных длин мы пока не обнаружили. И теперь будем по очереди обрабатывать элементы массива, обновляя эти значения. Заметим два важных свойства:

- $d[i - 1] \leq d[i]$ для всех $1 \leq i \leq n$;
- каждый элемент h_i обновляет максимум один элемент массива \mathbf{d} — после элементов, больших h_i , поставить h_i нельзя, а если можно поставить h_i на k -е место, то уже существовала последовательность длины $k - 1$, заканчивающаяся на что-то, меньшее h_i .

Это означает, что при обработке очередного h_i , мы можем за $\mathcal{O}(\log n)$ с помощью двоичного поиска в массиве \mathbf{d} найти первое число, которое больше либо равно текущему h_i , и обновить его. Отдельно стоит уделить внимание восстановлению ответа, то есть последовательности прямоугольников, которые необходимо использовать — для этого достаточно просто в момент обновления \mathbf{d} запоминать для нового элемента стоящий перед ним. Время работы — $\mathcal{O}(n \log n)$.