Разбор задачи «Продвижение»

Вероятно, это самая простая задача комплекта. Перебирая в цикле числа a_i (сохранять их было не обязательно), нужно было либо прибавлять очередное число к b (если $a_i < m$) или же прибавлять единицу к f (если $a_i \geqslant m$). После того, как все n чисел a_i были обработаны, оставалось лишь вывести f и b.

Разбор задачи «Эмблема»

Все, что требовалось в этой задаче — аккуратно обработать входные данные. При этом значения $minf$ и $mins$ получаются суммированием «гарантированных» голосов — тех, которые совершенно точно будут поданы за соответствующий проект. Если же голос может быть как подан за некоторый проект, так и не подан, то его следует учесть в значениях $maxf$ и / или $maxs$ (соответственно).

Разбор задачи «Экскурсии»

Количество площадок и количество запросов относительно невелико, и это позволяет использовать практически любую подходящую структуру данных. Например, можно поддерживать список, в котором хранятся списки отрезков маршрутов. Отрезок маршрута задается первым и последним проходом, которые в него входят. В один список при этом объединяются отрезки, имеющие одинаковую длину. А эта длина, в свою очередь, является индексом списка списков.

Изначально в этом списке списков всего один элемент — отрезок, совпадающий с полным исходным маршрутом. Когда приходит запрос на объявление некоторого прохода небезопасным, нужно найти отрезок, в котором содержится этот проход, удалить этот отрезок из того списка, в котором он в настоящий момент находится, и сформировать один или два новых отрезка.

Понятно, что один отрезок будет сформирован, если объявленный небезопасным проход был первым или последним проходом в своем отрезке. Если же небезопасный проход был единственным проходом в своем отрезке, новый отрезок формироваться не будет. Если же небезопасный проход не был ни первым, ни последним в своем отрезке, то будут сформированы два отрезка. После формирования отрезка (или отрезков) его следует поместить в соответствующий его длине список.

Когда приходит запрос на объявление некоторого прохода безопасным, также возможно три варианта: этот проход соединит два отрезка маршрута, этот проход удлинит на единицу один отрезок маршрута, этот проход образует новый отрезок, состоящий из этого прохода.

Наконец, когда приходит запрос на максимально возможную длину маршрута, нужно выяснить, какой максимальной длине соответствует непустой список (учитывая ограничения, достаточно просмотреть списки для всех длин).

Еще один вариант решения может быть таким.

Ограничения в данной задаче позволяют как выполнять запросы изменения, так и отвечать на запросы максимальной длины за $O(n)$ на запрос.

Заведем массив $ends$, где $ends[i] = j$ $(i \leqslant j)$, если в процессе изменений образовался выделенный маршрут от i до j; -1 в ином случае. Отметим, что площадка всегда лежит на каком-либо маршруте — даже если этот маршрут начинается и заканчивается в ней.

Теперь рассмотрим выполнение запросов изменения.

- Разрушение прохода $i \to i + 1$.

  Пусть площадка i принадлежит маршруту $[start, end]$. Тогда после разрушения образуются два отрезка $[start, i]$ и $[i + 1, end]$.

  Найти отрезок просто — найдем в цикле площадку, для которой $j \leqslant i \leqslant ends[j]$. После этого проведем две операции присваивания:

  $ends[i + 1] = ends[j]$

  $ends[j] = i$

- Восстановление прохода $i \to i + 1$

Найдем начало маршрута, концом которого является площадка i (она точно является концом какого-либо маршрута, так как проход после нее был разрушен). Обозначим стартовую площадку j.

Восстановление маршрута соединяет маршруты $[j, i]$ и $[i+1, ends[i+1]]$. Объединим их за два присваивания:

$ends[j] = ends[i+1]$

$ends[i+1] = -1$

- Нахождение максимальной длины маршрута

  Данный запрос выполняется проще запросов изменения. Ответом является максимум среди величин $ends[i]$, где i пробегает по всем площадкам — максимум также можно найти проходом по массиву.

Разбор задачи «По законам жанра»

Эта задача решается методом динамического программирования.

Обратим внимание на то, что во входных данных материалы уже даны в хронологическом порядке, т.е. если у одного материала номер меньше, чем у другого, это означает что он снят раньше.

Обозначим через $dp[i]$ максимальную суммарную интересность последовательности материалов, которую можно получить, если материал $\#i$ будет показан последним.

Переход к $dp[i+1]$ осуществляется следующим образом: среди всех $dp[k]$ $(k = 0, 1, \ldots, i)$ найдем максимальное значение среди тех, для которых верно $a_k + w_k < a_{i+1} + w_{i+1}$. Обратите внимание, что неравенство должно быть строгим, поскольку показывать два материала в один день нельзя: их порядок в этом случае не будет определен.

Таким образом $dp[i+1] = \left(\max\limits_{k=0}^{i} dp[k] | (a_k + w_k) < (a_{i+1} + w_{i+1})\right) + c_{i+1}$, а выбранное k_{max} будет номером материала, предшествующего материалу $\#(i+1)$.

Когда все значения $dp[i]$ будут вычислены, максимальное из них и даст нам максимальную суммарную интересность показанных материалов. Чтобы получить список материалов, необходимо завести еще один массив, в котором для каждого материала будем хранить номер его непосредственного предшественника или 0 (или другой «несуществующий номер»), если такового нет. Т.е. для материала $\#(i+1)$ $parent[i+1] = k_{max}$.

Пусть i_{max} — индекс максимального значения среди всех $dp[i]$. Тогда получить список показанных материалов можно обратным проходом по массиву $parent[]$, начиная с $parent[i_{max}]$. Если при формировании списка элементы добавляются в его конец, его нужно будет выводить в обратном порядке.

Асимптотика описанного решения — $O(n^2)$. Можно написать решение и с более хорошей асимптотикой.