Задача A. Матч за звание чемпиона

Автор: Никита Небабин
Разработчик: Григорий Базилевич

Для решения этой задачи воспользуемся фактом, описанным в этой статье: https://ru.wikipedia.org/wiki/Король_и_пешка_против_короля

Если король находится в квадрате пешки, то ответ «NO», иначе «YES». Значит, для решения тестового случая нам необходимо понять, находится ли король в квадрате пешки, или нет. Обозначим длину квадрата пешки за $d = n - r_p$. Тогда король находится в квадрате пешки, если выполняются такие условия:

$$\begin{cases} r_p \leqslant r_k \\ c_p - d < c_k \\ c_k < c_p + d \end{cases}$$

Итоговая асимптотика: $O(t)$.

Задача B. Путешествие во времени

Автор и разработчик: Матвей Кориненко

Давайте переформулируем задачу. СМП представляет из себя ориентированный граф, где вершинами являются годы, а ребрами — Межвременные Переходы. Ответом является 0 только в случаях, если из года, в котором живет Валера (X) не существует перехода в другие года, либо если по пути из года X можно попасть в цикл, то есть остаться в СМП навсегда. В остальных случаях ответом является разность между X и максимальным годом, в который может переместиться Валера (то есть будет выиграно минимально возможное время на подготовку).

Подзадачи 1, 2 Для решения этих подзадач можно написать обход в глубину, использующий три состояния вершин: вершина v не посещена ни разу (0), из вершины v был запущен обход в глубину (1), из вершины v был запущен обход в глубину и пройдены все смежные с ней вершины (2).

Если мы в течение обхода попадаем в вершину v с состоянием 0, мы меняем его на состояние 1 и продолжаем обход из вершин, смежных с v.

Если же мы попадаем в вершину с состоянием 1, то значит в нашем дереве обхода в глубину есть обратное ребро в эту вершину, а значит есть цикл во всем графе. Тогда обход можно прерывать и выводить 0.

Если вершина v имеет состояние 2, то мы ее игнорируем и идем дальше, потому что это значит, что из поддерева v в дереве обхода в глубину нет обратных ребер в нее, но попасть в нее можно несколькими способами. Уже имеющийся ответ это не меняет никак.

Подзадача 3 Для решения этой подзадачи можно просто найти самый большой год, смежный с годом X и вывести разность с ним. Для этой и следующих подзадач значения годов могут существенно превышать 10^8, поэтому нужно переприсвоить годам новые индексы или использовать *map* или *unordered map*.

Подзадача 4 Для решения этой подзадачи можно проигнорировать проверку на циклы и просто обходом в глубину пройтись по всему графу.

Подзадача 5 Для решения этой подзадачи необходимо использовать идею решения подзадач 3 и 4 и оптимизировать хранение вершин ранее описанными методами.

Задача C. Солнечные лучи

Автор: Ольга Поспелова
Разработчики: Ольга Поспелова, Матвей Кориненко

Подзадачи 1 — 4 Данные подзадачи можно было решить решением за $O(nk)$.

Для каждого отрезка длины k будем за проход по массиву находить минимальное показание *«SolarLightMetr»*, потом выберем из полученных $n - k + 1$ значений наибольшее.

Подзадача 5 Для решения данной подзадачи воспользуемся идеей амортизированной линии. Этот подход можно кратко описать так: каждый элемент один раз добавится в какую-то структуру и один раз удалится из нее.

Теперь к реализации. Создадим мультисет, в нем будем поддерживать значения на отрезке от $[max(0, i - k + 1); i]$. Теперь пройдем за $O(n)$ по массиву, будем добавлять i-й элемент, удалять, если надо, $(i - k)$-й. Тогда для отрезка мы умеем находить ответ за $O(1)$ — взять первый элемент в мультисете. Переход от i к $(i + 1)$ за $O(\log n)$, получаем решение за $O(n \log n)$, которое успешно проходит 5 подзадачу.

Подзадача 6 Для решения 6 подзадачи улучшим идею, описанную выше, убрав лишний $\log n$ из асимптотики. Вместо мультисета возьмем дек, в котором будем поддерживать индексы позиций (по возрастанию индексов) так, чтобы при проходе по деку значения под рассматриваемыми индексами шли по неубыванию.

Переход от i к $(i + 1)$ элементу:

1. Удалить первый элемент дека, если он равен $i - k$. (Т.к. этот элемент уже не попадает в рассматриваемый отрезок)
2. Пока дек имеет элементы и значение в массиве под индексом, хранящимся в деке на последней позиции, больше, чем значение на i-й позиции в массиве, удалим один элемент с конца. (Для достижения условия, описанного выше)
3. Добавим i в конец дека.
4. Пересчитаем ответ для отрезка $[i - k + 1; i]$ (если надо). Минимальным будет элемент на позиции, хранящейся в начале дека.

Все эти операции дек умеет выполнять за $O(1)$, значит мы смогли убрать лишний $\log n$ из асимптотики.

Итоговая асимптотика: $O(n)$.

Задача D. Орлиная высота

Автор и разработчик: Матвей Соловьёв

По условию, $w_{i+1} = w_i \times c_{i+1}$. Тогда высота в первый день равна $w_1 = w_0 \times c_0 = w_0 \times \frac{p_0}{q_0}$. Аналогично посчитаем все высоты до n-го дня. Получим высоту в n-й день равной $w_n = w_{n-1} \times c_n = w_{n-1} \times \frac{p_n}{q_n} = w_{n-2} \times \frac{p_{n-1} \times p_n}{q_{n-1} \times q_n} = w_i \times \frac{\prod_{j=0}^{n-i-1} p_{n-j}}{\prod_{j=0}^{n-i-1} q_{n-j}} = w_0 \times \frac{\prod_{j=1}^{n} p_j}{\prod_{j=1}^{n} q_j}$. Мы получили отношение высоты в последний день к высоте в нулевой день: $\frac{w_n}{w_0} = \frac{\prod_{j=1}^{n} p_j}{\prod_{j=1}^{n} q_j}(1)$. Тогда, чтобы получить минимально возможное значение w_0, можно посчитать данные произведения и сократить полученную дробь, т. е . $Q = \frac{\prod_{j=1}^{n} p_j}{gcd(\prod_{j=1}^{n} p_j, \prod_{j=1}^{n} q_j)}(2)$. Аналогично минимально возможное w_n будет равно $P = \frac{\prod_{j=1}^{n} q_j}{gcd(\prod_{j=1}^{n} p_j, \prod_{j=1}^{n} q_j)}(3)$. Так как высоты в каждый день не превышают 10^{18} метров, то,

подсчитывая числитель и знаменатель дроби по порядку и сокращая их на каждом шаге на максимально возможное значение, то есть на их НОД, позволит не выйти за пределы целочисленного типа.

Подзадача 1 Так как в данной подзадаче ограничения на L и R не превышают 10^6, то можно проверять каждую высоту от L как высоту в нулевой день, пока высота в последний день соответствует неравенству $w_n < R$ для данного w_0. Подсчитывать w_n можно с помощью равенства (1).

Подзадача 2 Для решения данной подзадачи будем считать ответ за $O(1)$. Посчитаем такое число k_{min}, что $L \leqslant k_{min} \times Q = k_{min} \times \frac{\prod_{j=1}^{n} p_j}{gcd(\prod_{j=1}^{n} p_j, \prod_{j=1}^{n} q_j)}$, учитывая равенство (2), при этом k_{min} минимально возможное. Также посчитаем k_{max} такое, что $R > k_{max} \times P = k_{max} \times \frac{\prod_{j=1}^{n} q_j}{gcd(\prod_{j=1}^{n} p_j, \prod_{j=1}^{n} q_j)}$, учитывая равенство (3), при этом k_{max} максимально возможное. Тогда ответом будет $max(k_{max} - k_{min} + 1, 0)$

Итоговая асимптотика: $O(n)$.

Задача E. Алиса и букет цветов

Автор: Иван Девятериков
Разработчики: Иван Девятериков, Григорий Базилевич, Матвей Кориненко

Подзадача 1 Для решения первой подзадачи можно для каждого из q запросов за $O(n)$ искать ответ.

Подзадача 2 (решение 1) Воспользуемся идеей сканирующей прямой, пройдемся ей по типам цветов. Рассмотрим запросы трех видов:

1. Тип -1 — начиная с этого типа цветы продаются в каком-то магазине m.
2. Тип 0 — необходимо найти ближайший из магазинов
3. Тип 1 — в магазине m цветы других типов не продаются.

Тогда нам необходимо добавить для каждого магазина тройки $(l_i;\ -1;\ i)$ и $(r_i;\ 1;\ i)$, а для каждого запроса тройку $(k_i;\ 0; i)$.

Теперь отсортируем эти тройки, пройдемся по ним. Если встречаем запрос типа -1, то мы добавляем магазин на позиции в set. Если встречаем запрос типа 1, то удаляем магазин из струкутры set. Встречая запрос типа 0, нам необходимо лишь посмотреть на первый элемент структуры set и записать его как ответ для данного запроса (либо сказать, что ответ -1).

Асимптотика: $O(n \log n)$.

Подзадача 2 (решение 2) Решим задачу в offline. Сохраним все запросы, отсортируем по типу цветов. Линейно пройдем по всем магазинам, с помощью бинарного поиска найдем запросы, которым удовлетворяет рассматриваемый магазин. Установим ответ за линейный проход по удовлетворяющим запросам, потом удалим их из сортированного массива. Для этого можно использовать структуру данных set. Асимптотика: $O(n \log n)$.

Задача F. Игра с колодами карт

Автор: Никита Поздняков
Разработчики: Никита Поздняков, Григорий Базилевич

Подзадача 1 Для решения данной подзадачи можно было перебрать все возможные порядки вытаскивания карточек.

Замечание Очевидно, что брать какую-то карточку с отрицательным значением нам выгодно тогда и только тогда, когда взяв ее и другие карточки, следующие за ней, мы сможем увеличить счет.

Подзадачи 2 — 3 Для решения данных подзадач можно было рассчитывать оптимальность от применения определенной карточки за проход по колоде.

Подзадачи 4 — 6 Давайте разобьем колоды на блоки. Назовем дельтой блока изменение счета от применения блока. Блок будет начинаться с отрицательного значения (или начала колоды), далее дельта блока будет как-то меняться и в какой-то момент времени она может стать положительной. Если такое произошло, то следующее отрицательное число, которое мы встретим на карточке, будет начинать новый блок. Так же необходимо заметить, что раз счет не может опускаться ниже нуля, то для применения какого-то блока нам необходимо иметь счет, больше либо равный минимальной дельты, достигаемой в блоке (минимальной дельты на префиксе блока).

Теперь мы заменим колоду последовательностью блоков, для каждого из которых мы знаем на сколько изменится счет при применении данного блока и какой минимальный счет для этого необходим. Если дельта отрицательная (что возможно только для последнего блока), то просто удалим этот блок из рассмотрения, тк он не улучшит нам счет.

Для решения 4 подзадачи мы можем выбирать блок, который будет применен на данном шаге за проход по всем колодам.

Для решения 5 и 6 подзадач необходимо было сложить значения первых блоков каждой колоды в какую-нибудь сортированную структуру, например, set. (Отстортируем по минимальному необходимому счету) Тогда применять мы будем блок, который находится на первой позиции в данной структуре. Если мы не можем его применить — мы не можем применить и блоки, стоящие далее. Когда мы применили блок, мы должны удалить рассмотренный блок и добавить следующий блок из данной колоды в структуру.

Данное решение проходит 5 и 6 подзадачи в зависимости от реализации.

Итоговая асимптотика: $O(\sum k \log(\sum k))$.