Большой потоп

Автор задачи: Константин Бац, разработчик: Мария Жогова

В задаче требовалось выбрать порядок подрыва башен такой, что сумма объемов воды, находящихся в башнях в момент взрыва, была максимальна.

Заметим, что моменты для разрушения башен из системы i желательно выбирать до t_i. Иначе, вода находящаяся в не взорванных башнях будет спущена, и это не увеличит искомую сумму.

Для прохождения тестов первой подзадачи можно было явно перебрать моменты для подрыва башен. Поскольку $k = 1$ и $t_i \leqslant 5$, в каждый момент можно было взорвать не более одной башни, поэтому для каждой секунды от 1 до 5 нужно выбрать, какую башню стоит взорвать в эту секунду. Перебором среди всех таких комбинаций можно найти такую, которая даёт максимальный результат.

Во второй подзадаче гарантировалось, что $b_i = 1$ и всё t_i разные. На самом деле, это означало, что единственную башню каждой подсистемы нужно взорвать за секунду до спуска воды, то есть на $t_i - 1$ секунде. Очевидно, что таком случае ответ будет максимальным.

Третья подзадача выделялась тем, что водосброс происходил во всех системах одновременно, пусть это будет секунда T. Тогда очевидно, что всего мы можем взорвать не больше $k \cdot T$ башен. Предположим, суммарно в системах больше $k \cdot T$ башен. Тогда выгоднее взрывать башни из систем с большим начальным уровнем воды. Действительно, давайте заменим одну башню из системы i на башню из системы j и $a_i > a_j$, тогда вне зависимости от момента взрыва из башни выльется строго меньше воды. Предположим у нас меньше $k \cdot T$ башен тогда выгодно начать взрывать башни как можно позже. При этом, если все башни в итоге будут взорваны, порядок разрушения не важен.

В четвёртой подзадаче в каждой системе была всего одна башня. Давайте объединим все ранее приведённые идеи. Рассмотрим системы с самым поздним сбросом воды, то есть максимальным $t_i = T$. Выберем из них k с максимальным a_i, очевидно, их выгодно разрушить в секунду $t_i - 1$. Перейдем к секунде $T - 1$. Добавим в рассмотрение системы с $t_i = T - 1$. Среди оставшихся с предыдущего шага и добавленных в рассмотрение снова выберем k систем с максимальными a_i. Будем продолжать, пока не дойдем секунды 0 или в рассмотрении не останется систем. Если в рассмотрении закончились системы, опять найдём среди оставшихся $\max t_i$ и повторим всё выше описанное. Хотя $t_i \leqslant 10^9$, будет всего $\mathcal{O}(n)$ секунд, когда мы будем выбирать системы. Для быстрого выбора системы с максимальным a_i можно использовать приоритетную очередь с добавлением, удалением и поиском максимума за $\mathcal{O}(\log n)$. Время работы решения — $\mathcal{O}(n \log n)$.

В подзадаче пять не было ограничения на количество башен в системе, но t_i–е не превышали 10^5. Для решения можно было воспользоваться идеей из четвёртой подзадачи. Выберем систему с максимальным начальным уровнем, взорвем в текущую секунду как можно больше башен из этой системы. Понятно, что, если у нас есть несколько систем с одинаковыми a_i, не важно, в какой системе разрушать башни. Поскольку башен может быть очень много, возможно, придётся каждую секунду выбирать системы, в которых нужно будет разрушить башни. Всего нужно будет не больше n раз выбрать систему с максимальным a_i и при этом рассмотреть не больше $\max t_i$ секунд. Значит временная сложность такого решения — $\mathcal{O}((n + \max t_i) \log n)$.

Для прохождения тестов последней подзадачи нужно было немного оптимизировать предыдущее решение. Пусть в течении p секунд с T по $T+p-1$-ю не происходит спусков воды, и в $T+p-1$-ю секунду максимальный начальный уровень воды имела система j. Тогда в течении p секунд можно подрывать башни только из системы j, и это не испортит ответ. Из этого следует, что нам не обязательно рассматривать каждую секунду. Достаточно рассматривать те секунды, когда мы добавляем в рассмотрение новую систему или в какой-то системе заканчиваются башни. Между этими моментами каждую секунду мы будем либо разрушать башни в какой-то определённой системе, либо ничего не разрушать. Таких моментов будет не больше $2n$, значит время работы такого решения будет $\mathcal{O}(n \log n)$.