Монстры и мечи

Автор и разработчик задачи: Ренат Каримов

Подгруппа $k = 1$

Для решения подгруппы k=1 введем сначала новое обозначение: w_i = минимальная стоимость меча, при помощи которого можно убить i-го монстра. Если же для какого-то монстра такого меча нет, то ответ сразу NO.

В этой подгруппе мы должны убивать монстров подряд по одному, следовательно каждый раз нужно находить стоимость самого дешёвого меча, который может убить очередного монстра. Это и есть w_i. Тогда чтобы решить задачу нам нужно просто n раз проверить, что количество монет, которое у нас осталось больше или равно очередного w_i. Решение работает $O(n \log n + m \log m)$

Подгруппа $k = n$

Для решения подгруппы с $k = n$ нужно заметить, что при покупке нового меча нам всегда нужно выбирать меч с сильной больше, чем предыдущего. Это так потому что у нас нет ограничения на прочность меча. Тогда из этого факта следует замечание, что в оптимальном решении мы всегда можем использовать меч до упора. То есть нужно менять меч только в ситуации, когда он не может убить очередного монстра.

Используя эти факты можно написать решение с использованием техники динамическое программирование. Обозначим $dp[i]$ = максимальное количество монет, которые может остаться у рыцаря после убийства первых i монстров (при этом нам не важно, какими мечами мы убивали монстров).

Для удобства обозначим $w(i, j) = \max\limits_{i \leqslant l \leqslant j} w[l]$. Тогда изначальное состояние: $dp[0] = x$ и переход $dp[i] = \max\limits_{j} dp[j] - w(i, j - 1) + r(i, j - 1)$. При этом должен быть верно, что $dp[j] \geqslant w(i, j - 1)$.

Для оптимизации такой дпшки можно заметить факт, что нас интересуют не все состояния этой дпшки. Для каждого меча можно выделить префикс монстров, на котором он может всех победить. Тогда на основе фактов выше можно пересчитывать дпшки только через эти префиксы, так мы получим решение за $O(m^2 + n \log n)$.

Для полного решения можно нужно просто оптимизировать динамику выше, это можно сделать, например, при помощи структуры декартово дерево, тогда итоговое решение будет работать за $O(n \log n + m \log m)$

Подгруппа k - произвольное

Для решения подгруппы k - произвольное нужно подробнее рассмотреть получившуюся формулу. Пусть, значение i - фиксированно, тогда уменьшая j значение $w(i, j)$ будет увеличиваться. Из-за этого хочется использовать технику стека максимумов. При помощи нее можно реализовать такую же дпшку как и для случая k=n только ещё с условием, что $i - j \leqslant k$. Так можно получить решения за $O(n \log n + m \log m)$ при помощи ДД, ХЛД или прохода с сетом.

Однако подробнее мы разберем немного другую дпшку. Обозначим $dp[i]$ = минимальное количество золота, которое нам нужно, чтобы гарантированно убить всех монстров с номерами i ... n-1 (в порядке возрастания номеров). Тогда ответ YES только в случае, если $dp[0] \leqslant x$.

Изначально значение: $dp[n-1] = w[n-1]$. Переход: $dp[i] = \min\limits_{i < j \leqslant i+k} (w(i, j-1) + \max(0, dp[j] - r(i, j-1)))$. Обозначим $pr[i] = r(0, i)$ и немного перепишем получившуюся формулу, получим $w(i, j - 1) + max(0, dp[j] - pr[j - 1] + pr[i])$. Тогда нам интересен знак $dp[j] - pr[j - 1] + pr[i]$. При этом несложно заметить, что если теперь уже j - фиксированно, а i уменьшается, то значение $dp[j] - pr[j - 1] + pr[i]$ только убывает, следовательно знак меняется не более одного раза.

Такое дп можно быстро пересчитывать при помощи стека максимумов и сетов, заметив факт, что $dp[i] - pr[i - 1] \geqslant dp[i + 1] - pr[i]$.

Итоговое решение будем иметь ассимптотику $O((n + m) \log (n + m))$