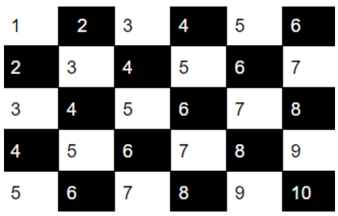
Задача A. Жонглер грибами

Так как грибы двигаются по окружности, то после n-го гриба появляется 1-й гриб, а значит момент времени n+1 соответствует грибу 1, также как и моменту времени $2 \cdot n + 1$, $3 \cdot n + 1$ и так далее. В общем случае моменту времени $t = m \cdot n + k$ соответствует гриб k, а если k = 0, то это гриб n. Для данной задачи нам достаточно хранить массив исходных символов и на каждой запрос t выводить элемент с индексом n, если t mod n = 0, а иначе выводить элемент с индексом t mod n.

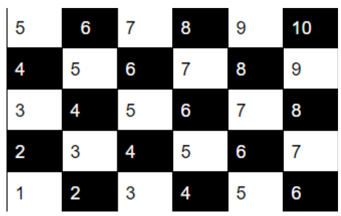
Задача B. Игра на шахматной доске

Если Йоукахайнен и Кузя оба находятся на диагоналях не того цвета, как клетка цели, то они никогда не смогут достичь цели, т.е. ответ – «Not interesting». Если только один из игроков на диагонали того же цвета, что и цель, то он является победителем. Далее рассмотрим решения на разные количества баллов:

- Решение на 30 баллов. Можно рассмотреть все случаи расположения игроков, которые могут быть в тестах (без учета случаев выше) и вручную подсчитать ответ для каждого из них.
- Решение на 60 баллов. Представим шахматную доску как граф, где вершины соответствуют клеткам шахматной доски, а ребра соответствуют возможных ходам игроков. Запустим от клетки-цели обход «в ширину» и будем поддерживать для каждой клетки расстояние от клетки-цели (расстояние клетки-цели до самой себя равно 0). Имея расстояния до клетки-цели мы можем решить, кто выиграет (или это будет ничья, если расстояния одинаковые).
- Решение на 100 баллов. Давайте изменим систему координат, теперь вместо номера строки и столбца, каждая клетка характеризуется номерами двух диагоналей, одна из которых параллельна главной диагонали, а другая параллельна побочной. Примеры:



Пример нумерации диагоналей параллельных побочной диагонали



Пример нумерации диагоналей параллельных главной диагонали

Формула номера диагонали для картинки слева: $\hat{x} = x + y - 1$, формула номера диагонали для картинки справа: $\hat{y} = y - x + n$. Теперь у каждой клетки шахматного поля появились новые координаты $\hat{x}$ и $\hat{y}$, в которых расстояние между какими либо клетками $(\hat{x_1}, \hat{y_1})$ и $(\hat{x_2}, \hat{y_2})$ – это $|\hat{x_1} - \hat{x_2}| + |\hat{y_1} - \hat{y_2}|$ – манхэттенское расстояние. В новых координатах мы можем найти расстояние до цели у каждого из игроков и сравнить их.

Задача C. Лестничный контест

Во-первых, если сложности не упорядочены на входе, то тогда необходимо отсортировать сложности по возрастанию. Во-вторых, если у нас уже взята какая-то задача t, то нам выгодно взять следующей задачу m, такую что разница между сложностями t и m была бы максимальная, и не превосходила бы k (так мы сможем взять как можно меньше задач, не нарушая условий). Таким образом, если в списке отсортированных по возрастанию сложности задач у каких-нибудь двух соседних задач разность сложности превосходит k, то ответа не существует и мы выводим -1. Иначе давайте будем жадно строить список задач, который мы взяли. Обязательно включим в начало списка 'a'. Будем хранить количество взятых задач и последнюю взятую задачу. Пройдем от задачи с наименьшей сложностью к задачу с наибольшей и каждый раз, когда разница сложностей между последней взятой задачей и текущей будет превосходить k,тогда мы будем брать предыдущую задачу и обновлять переменные количества взятых задач и последней взятой задачи. Если по окончании процесса, задача b не была взяла, тогда также включаем ее в список.

Задача D. Марюшки

Рассмотрим решения на разные количества баллов

- Решение на 20 баллов. Проверить подходит ли единственная данная нам марьюшка под ограничения – если да, то вывести ее, иначе – вывести 0.

- Решение на 50 баллов. Один из вариантов решения – отсортировать марьюшки по возрастанию времени, за которое они съедаются и во вторую очередь по радости, которую они приносят. Теперь для каждой марьюшки, найдем бинарным поиском (правосторонним) вторую подходящую под ограничения марьюшку. Если не существует второй подходящей под ограничения марьюшки, тогда учитываем вариант только с одной. Из всех таких вариантов выбираем в ответ вариант с максимальной суммарной радостью.

- Решение на 100 баллов. Будем поддерживать такую динамику – $dp[n][m][t]$ – максимальная радость, которую могут принести m марьюшек с суммарными временем t из рассмотренного списка из первых n марьюшек. Для каждой марьюшки i – мы можем, либо включить ее в список взятых, либо не включить, тогда переходы динамики такие: $dp[i][m][t] = max(dp[i-1][m][t], dp[i-1][m-m_i][t-t_i] + s_i)$. Так как $1 \leqslant n, m, t \leqslant 10^5$, то размеры массивов и пересчет динамики будут относительно небольшими. Также будем поддерживать массив cnt с теми же параметрами, но значение которого будет не наибольшая радость, а последняя марьюшка, которая принесла наибольшую радость (используется для восстановления динамики).

Задача E. Путь домой

- Решение на 50 баллов. Давайте найдем какой-либо один путь из a в b, например обходом 'в глубину'. Теперь попробуем удалить по отдельности каждое из ребер и найти путь из a в b.Если при удалении каждого из ребер мы ни разу не нашли пути из a в b, то ответ «NO», иначе «YES».

- Решение на 100 баллов. Дадим каждому ребру графа вес 0, кроме ребер на привычном пути – их вес будет 1. Если длина кратчайшего пути от в b будет равна k, то ответ «NO>, иначе – «YES». Теперь кратчайшее расстояние можно найти 0-1 BFS-ом.