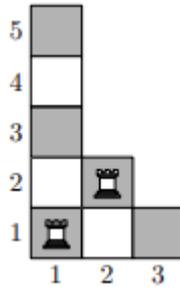


Задача А. Шахматы

Каждая ладья бьёт строку и столбец, но не более. После этого понятно, что нам выгодно выставлять ладьи на диагонали, которая выходит из левого нижнего угла.



Задача В. Дюны

Порывы ветра:

- было n порывов ветра
- i -ый порыв менял высоту участков дюны с l_i -го по r_i -й
- изменение высоты: $+x_i, -x_i, +x_i, -x_i, \dots$

Запросы:

- m запросов: «какая высота в итоге у q_i -го участка?»

Количество участков от 1 до миллиарда, поэтому не надо моделировать миллиард участков дюны!

Для каждого запроса – участка номер q_i – вычислим его высоту. Для этого просмотрим заново все n порывов ветра, про каждый посчитаем, менял ли он высоту участка q_i и как именно.

Задача С. Игра

Мальчикам выгодно начать кидать на столбики, которые есть у обоих.

Считаем both — количество столбиков, на которые оба могут закинуть.

N_{Petya} и N_{Vasya} — количества столбиков, на которые может закинуть только Петя и на которые может закинуть только Вася, соответственно.

Далее моделируем ходы по очереди.

- both > 0 , то обязательно закидываем на общие.
- both = 0 и $N_{Player} > 0$, то закидываем на соответствующий столбик.

- both = 0 и NPlayer = 0, то больше не закинуть.

Задача D. Мерлин

Заметим, что выгоднее всего выливать эликсир из тех сосудов, в котором его больше всего.

Отсортируем массив сосудов по возрастанию (нужно использовать одну из быстрых сортировок, например quick sort, merge sort, heap sort, этого требуют ограничения).

Если мы выльем k последних (в отсортированном порядке) сосудов, то останется $n-k$ первых. Чтобы в $n-k$ первых сосудов было поровну эликсира, нужно долить для каждого сосуда a_i ($1 \leq i \leq n-k$) хотя бы $a_{n-k} - a_i$ литров эликсира. Тогда на k накладывается следующее условие: $(a_{n-k} - a_1) + (a_{n-k} - a_2) + \dots + (a_{n-k} - a_{n-k}) \leq a_{n-k+1} + \dots + a_n$. Это выражение можно преобразовать:

$$a_{n-k} \cdot (n - k) - \sum_{i=1}^{n-k} a_i \leq \sum_{i=n-k+1}^n a_i$$

Теперь мы можем перебирать k , начиная с нуля до тех пор, пока это условие не выполнится.

Ограничения на n не позволят каждый раз считать эти суммы заново, поэтому просто предподсчитаем суммы элементов на всех префиксах и суффиксах массива.

Также нужно обратить внимание, что эти суммы могут не поместиться в 32-битный тип данных, и использовать 64-битный.