Задача B. Цветы

Наименьшее количество букетов равняется одному при нечётном n и двум при чётном n (первый букет — из трёх цветов, второй — из $n-3$ цветов).

Чтобы получилось наибольшее количество букетов, нужно составить как можно больше букетов из трёх цветов. Если n делится на 3, то можно сделать $\frac{n}{3}$ букетов. Если n даёт остаток 1 при делении на 3, то можно сделать $\frac{n-4}{3}$ букетов (в одном — семь цветов, во всех остальных — по три цветка). Если n даёт остаток 2 при делении на 3, то можно сделать $\frac{n-2}{3}$ букетов (в одном — пять цветов, во всех остальных — по три цветка).

Задача C. Сломанный Shift

В этой задаче нужно просто перебрать все пары соседних букв в строке, добавляя к ответу единицу, если регистр этих букв различается. И, если первая буква в строке заглавная, нужно добавить к ответу ещё единицу.

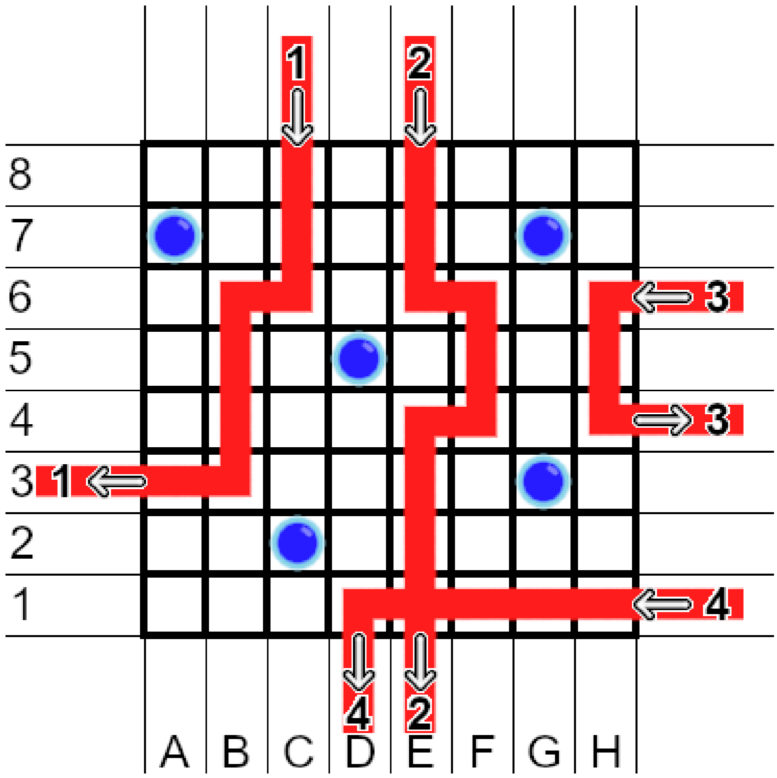
Задача D. Бедная Полина

Оказывается, если для какого-то i $a_i \neq i$, то Полина не сможет выполнить все дела.

Понятно, что Полина не сможет выполнить дело, если оно находится в цикле с другими делами. Поэтому покажем, если для какого-то i $a_i \neq i$, то i-е дело находится в цикле. Поскольку $a_i \neq i$ и в списке a присутствуют все числа от 1 до n, то есть индекс $j \neq i$, у которого $a_j = i$. Очевидно, тогда $a_j \neq j$. Тогда, по той же логике, есть индекс k, у которого $a_k = j$. Если $k = i$, то мы нашли цикл. Иначе продолжим рассуждения для k и так далее. Так как список конечен, рано или поздно мы попадём на индекс i, т.е. найдём цикл.

Задача E. Чёрный ящик

У данной головоломки ровно одно решение:



Догадаться до него можно так:

1. Очевидно, что луч 4 не может подняться вверх. Значит в клетке `C2` гарантированно есть шар.
2. Луч 1 не может идти по столбцу `A`, так как он не может повернуть в него. Значит луч 1 проходит по столбцу `B` и поворачивает в клетке `B3`. Можно заключить, что в одной из клеток `A5`, `A6`, `A7` или `A8` есть шар, который повернёт луч 1 в столбец `B`.

3. Луч 1 поворачивает нечётное число раз, так как он входит по вертикали, а выходит по горизонтали; и мы знаем, что как минимум два раза. Если луч 1 поворачивает пять или более раз, он должен касаться всех шаров. Можно проверить, что такое расположение не может быть решением. Значит, луч 1 поворачивает три раза. Для этого в одной из клеток `D3`, `D4`, `D5` или `D6` должен быть шар.

4. Тогда луч 2 не может пройти всё поле насквозь. Он должен повернуть хотя бы четыре раза, и нам осталось поставить только два шара. Нужную траекторию можно получить, если поставить два шара в столбец `G` или `H`, тогда луч 2 коснётся шара в столбце `D` дважды.

5. У нас осталось очень мало возможных расположений шаров. Есть только одно, которое подходит лучу 3.

Задача F. Гороскоп

Если для каждого предпочтения перебирать всех парней и проверять, подходят ли они под него, такое решение будет работать за время $O(n \cdot m)$ и не уложится в заданное ограничение по времени.

Давайте посчитаем для каждого месяца (от 1 до 12) значение $M[i]$ — сколько парней родилось в i-й месяц. Аналогично, для каждого дня (от 1 до 31) посчитаем значение $D[i]$ — сколько парней родилось в i-й день месяца, и для каждой возможной пары день-месяц посчитаем значение $DM[i][j]$ — сколько парней родилось в i-й день j-го месяца (при этом неважно, бывает ли в j-м месяце столько дней — если не бывает, то мы просто получим 0 парней). Тогда ответом для предпочтения $(a,\ b)$ будет являться значение величины $D[a] + M[b] - DM[a][b]$.

Задача G. Золотая Рыбка

Допустим, мы знаем, что i-е и j-е желания должны идти друг за другом. Давайте определим, какое из них идёт первым, а какое — вторым.

Пусть до выполнения этих двух желаний у нас уже накопилось x рублей. Если сначала идёт i-е желание, потом j-е, то после них у нас будет

$$(a_i \cdot x + b_i) \cdot a_j + b_j$$

рублей. А если идёт сначала j-е желание, потом i-е, то после них у нас будет

$$(a_j \cdot x + b_j) \cdot a_i + b_i$$

рублей. Так как все последующие желания увеличат количество рублей тем больше, чем больше рублей мы получим сейчас, то надо сравнить эти два значения и жадно выбрать тот вариант, который нам даёт больше денег.

Таким образом, i-е желание в этом случае идёт после j-го, если выполняется неравенство

$$(a_i \cdot x + b_i) \cdot a_j + b_j \leqslant (a_j \cdot x + b_j) \cdot a_i + b_i.$$

Раскрыв скобки, заметим, что в обеих частях есть слагаемое $a_i \cdot a_j \cdot x$. Избавимся от него и получим неравенство

$$b_i \cdot a_j + b_j \leqslant b_j \cdot a_i + b_i.$$

Это неравенство не зависит от x, значит порядок двух соседних желаний не зависит от текущего количества денег.

Более того, это неравенство можно упростить еще больше. Перенесём слагаемые b_j и b_i в противоположные стороны и получим

$$b_i \cdot (a_j - 1) \leqslant b_j \cdot (a_i - 1).$$

Наконец, перенесём множители b_i и b_j в противоположные стороны, сделав их знаменателями. Получим

$$\frac{a_j - 1}{b_j} \leqslant \frac{a_i - 1}{b_i}.$$

Из этого можно сделать вывод: раньше должно идти то желание, у которого значение $\frac{a_i-1}{b_i}$ меньше. Таким образом, задача свелась к сортировке по этому значению. Сравнение необходимо производить в целых числах, потому что при $a_i, b_i \leqslant 10^9$ две различные дроби могут отличаться на число порядка 10^{-18}, что превышает точность стандартных типов `float` и `double`. В Python можно использовать модуль `decimal` или `functools.cmp_to_key`.

Задача H. Охотники за привидениями

Для каждого луча найдём точку его пересечения со стеной, если такая есть. После этого переберём все пары лучей. Если какая-то пара лучей пересекается и для одного из них точка их пересечения находится ближе к охотнику, чем точка пересечения этого луча со стеной (или у этого луча нет пересечения со стеной), то ответ задачи: `YES`.

Будем обозначать векторное произведение векторов v_1 и v_2 как $v_1 \times v_2$. Векторное произведение — это произведение длин векторов на синус угла между ними. Для векторов $v_1 = (x_1, y_1)$, $v_2 = (x_2, y_2)$ выполняется $v_1 \times v_2 = x_1 \cdot y_2 - y_1 \cdot x_2$.

Жюри предлагает проверять, что лучи пересекаются, и находить точку их пересечения следующим способом. Пусть лучи заданы точками начала p_1 и p_2 и направляющими векторами v_1 и v_2 соответственно. Сначала проверим, что прямые, содержащие эти лучи, не параллельны — это выполняется тогда и только тогда, когда $v_1 \times v_2 \neq 0$. Если прямые не параллельны, то у них есть точка пересечения p.

Точка p лежит на первой прямой, поэтому её можно представить в параметрическом виде:

$$p = p_1 + v_1 \cdot t,$$

где t — пока неизвестный нам скаляр. Точка p также лежит на второй прямой, поэтому её можно представить в виде

$$p = p_2 + v_2 \cdot k,$$

где k — другой неизвестный нам скаляр. Получаем равенство

$$p_1 + v_1 \cdot t = p_2 + v_2 \cdot k.$$

Немного преобразуем его:

$$v_1 \cdot t = (p_2 - p_1) + v_2 \cdot k.$$

Теперь векторно умножим обе части равенства на v_2. Получим

$$(v_1 \times v_2) \cdot t = (p_2 - p_1) \times v_2 + (v_2 \times v_2) \cdot k.$$

Заметим, что мы вынесли скаляры t и k за скобки, и у нас получился множитель $v_2 \times v_2$. Он равен 0, так как вектор v_2 параллелен сам себе. Получаем

$$(v_1 \times v_2) \cdot t = (p_2 - p_1) \times v_2,$$

откуда легко выражается

$$t = \frac{(p_2 - p_1) \times v_2}{v_1 \times v_2}$$

Чтобы точка p лежала на первом луче, должно выполняться $t \geqslant 0$. Аналогично вычислим k по похожей формуле

$$k = \frac{(p_1 - p_2) \times v_1}{v_2 \times v_1}$$

и проверим, что $k \geqslant 0$. Так мы убедимся, что точка p является точкой пересечения не только прямых, но и лучей.

Также нужно будет найти пересечение лучей бластеров с отрезком AB. Эта задача сводится к поиску пересечения двух лучей, только нужно проверить, что луч бластера пересекается как с лучом AB, так и с лучом BA. После чего можно проверить, что либо точек пересечения нет вообще, либо самая ближняя — точка пересечения со стеной. На самом деле, сами точки в задаче можно не искать — достаточно найти соответствующие им параметры t и выполнить для них аналогичную проверку.