

Задача А. Проверка комнат в ГЗ

Для решения задачи необходима аккуратная реализация подсчета выведенных из эксплуатации комнат.

Для начала, научимся разбивать номер комнаты на этаж и комнату. Для этого проверим последний символ номера, если он 1 или г, то это комната в блоке и этаж это все кроме последних трех символов, иначе — это блок или одиночная комната, и этаж — все, кроме последних двух цифр. Зная номер этажа и номер комнаты, нетрудно определить, перед нами блок или комната.

Заведем словарь словарей, первым ключом будет номер этажа, вторым — номер комнаты (как строка с 1 или г, если имеется), а значением — количество человек, проживающих в комнате. Тогда для обработки комнаты в списке непригодных необходимо всего лишь занулить значение в нужной ячейке. Обработывая блок, необходимо занулить значения для левой и правой комнат.

Этажи 2, 17-18 требуют дополнительной обработки в начале (изначально занулить значения для комнат, которых нет).

Для получения ответа про этаж нужно всего лишь пробежать по значениям каждого элемента словаря этажа запроса и просуммировать их.

Задача В. Крысы Банаха - Тарского

Переформулируем условие. Дан граф-дерево, у которого пометили несколько вершин. Из какой-то вершины графа запускается обход. На нулевом шаге посещается исходная вершина, на первом шаге — все вершины на расстоянии 1 от стартовой, на k -м шаге — все вершины на расстоянии k . Шаг, на котором будет посещена хотя бы одна помеченная вершина, будет последним. Нужно узнать, сколько помеченных вершин будет посещено на этом шаге. Таких стартовых вершин тоже может быть несколько.

Для решения задачи предсчитаем для каждой вершины графа две величины: расстояние до ближайшей помеченной вершины и количество помеченных вершин на таком расстоянии. Для этого запустим обход в ширину сразу от всех помеченных вершин. Если на очередном шаге алгоритма сосед рассматриваемой вершины ещё не был помечен, помечаем его и обновляем для него ответ ($\text{dist}_{\text{neighbour}} = \text{dist}_{\text{current}} + 1$, $\text{num}_{\text{neighbour}} = \text{num}_{\text{current}}$). Если же соседняя вершина была помечена и расстояние до неё равно $\text{dist}_{\text{current}} + 1$, то увеличиваем счётчик помеченных вершин для соседней $\text{num}_{\text{neighbour}} += \text{num}_{\text{current}}$.

Далее по мере поступления запросов r_i достаточно будет выводить предсчитанный ответ для соответствующей вершины.

Задача С. Верстовые столбы

Для начала необходимо понять, как устроен граф. В условии сказано, что все рёбра можно ориентировать так, чтобы из каждой вершины выходило ровно одно ребро. Это означает, что каждая компонента связности графа является циклом, к каждой вершине которого подвешено дерево (возможно, состоящее из одной вершины).

Сначала научимся считать ответ для цикла длины L , где длина цикла — сумма длин всех входящих в него рёбер. Пусть мы зафиксировали цвет для первой вершины в цикле. Воспользуемся методом динамического программирования. Мысленно разобьём каждое ребро длины len на len рёбер длины 1, отметив все промежуточные вершины так, чтобы каждому верстовому столбу соответствовала вершина. Будем считать $\text{dp}[i][j]$ — количество способов покрасить в не более чем K цветов первые i вершин цикла так, чтобы любые две соседние вершины были разного цвета, а также при $j = 0$ последняя покрашенная вершина отличалась по цвету от первой вершины, а при $j = 1$ — совпадала. Ответ будет находиться в $\text{dp}[L][0]$. Ниже представлены формулы перехода:

$$\begin{cases} \text{dp}[i+1][0] &= \text{dp}[i][0] \cdot (K-2) + \text{dp}[i][1] \cdot (K-1) \\ \text{dp}[i+1][1] &= \text{dp}[i][0] \cdot 1 + \text{dp}[i][1] \cdot 0 \end{cases}$$

Для ускорения подсчёта ответа для цикла необходимо быстро возводить в степень матрицу перехода:

$$P = \begin{pmatrix} K-2 & K-1 \\ 1 & 0 \end{pmatrix}^{L-1}$$

$$\text{ans}_{\text{cycle}} = \begin{pmatrix} K \\ 0 \end{pmatrix} \cdot P \cdot \begin{pmatrix} 0 & 1 \end{pmatrix} = K \cdot P_{01}$$

После возведения в степень необходимо взять правый-верхний элемент матрицы и умножить его на K — количество способов зафиксировать цвет первой вершины.

Далее научимся добавлять к ответу дерево, отходящее от вершины цикла. Для этого заметим, что если начинать строить дерево от вершины из цикла (для которой уже зафиксирован цвет), то при добавлении ребра длины 1 есть ровно $K - 1$ способ покрасить новую вершину. Ребро длины len можно представить как len рёбер длины 1. Тогда ответ для компоненты связности будет считаться по формуле ниже (где $\sum |\text{edges}(\text{Tree})|$ — сумма длин всех рёбер в отходящих от цикла деревьях):

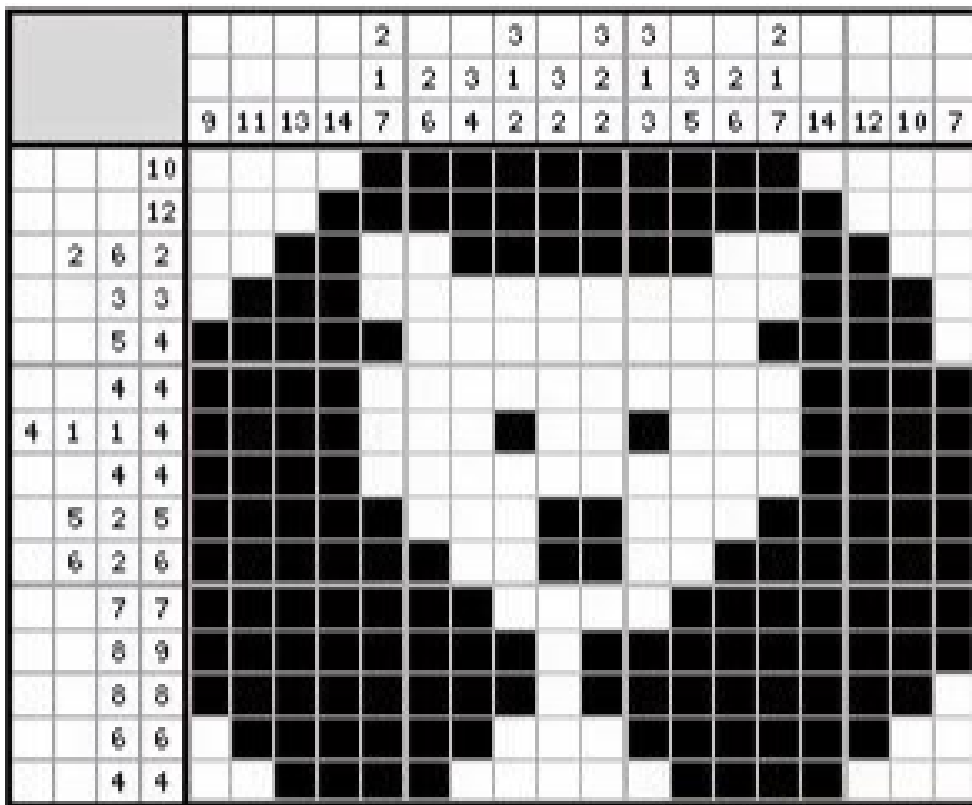
$$\text{ans}_{\text{comp}} = \text{ans}_{\text{cycle}} \cdot (K - 1)^{\sum |\text{edges}(\text{Tree})|}.$$

В конце необходимо перемножить результаты для всех компонент, так как каждую из них можно раскрашивать независимо.

Если программа с таким решением получала вердикт 'Превышен предел времени', то, например, можно добавить мемоизацию для вычисления степеней матрицы.

Задача D. Подарок из Японии

Давайте посмотрим на получившуюся после решения нонোগрамма картинку, можно догадаться, что это медведь и соответственно вывести «bear»:



Если же вам здесь видится другое животное и вы получаете вердикт неверный ответ, то можно вспомнить, что это Пермская олимпиада, а символ Пермского края это медведь.

Задача E. Путешествие по бамбуку

Сначала найдём центр отрезка (l, r) . Для этого с помощью бинарного поиска найдём наибольшее m такое, что запрос $(? \ 1 \ m)$ возвращает 0. Для этой вершины верно следующее неравенство: $m - 1 \leq (l - 1) + 1 + (r - m) \Leftrightarrow l + r \geq 2m - 1$. Так как для следующей вершины кратчайший путь проходит через добавленное ребро, выполняется следующее неравенство: $(m + 1) - 1 > (l - 1) + 1 + (r - (m + 1)) \Leftrightarrow l + r < 2m + 1$. Таким образом $l + r \in \{2m - 1, 2m\}$.

Вторым шагом найдём половину длины простого цикла, образованного добавленным ребром. Для этого снова запустим бинарный поиск и найдём наибольшее k такое, что запрос $(? \ m - \lceil \frac{k}{2} \rceil \ m + \lfloor \frac{k}{2} \rfloor)$ возвращает 0. Для k верно следующее неравенство:

$$m + \left\lfloor \frac{k}{2} \right\rfloor - m + \left\lceil \frac{k}{2} \right\rceil \leq \left(r - m - \left\lfloor \frac{k}{2} \right\rfloor \right) + 1 + \left(m - \left\lceil \frac{k}{2} \right\rceil - l \right) \Leftrightarrow 2k - 1 \leq r - l.$$

Аналогично для $k + 1$ верно неравенство: $2k + 1 > r - l$. Таким образом $r - l \in \{2k - 1, 2k\}$, а длина цикла — $2k$ или $2k + 1$. Соответственно, если $k = 1$, то единственный вариант ответа — $l = m - 1, r = m + 1$.

Заметим, что для $l + r$ и для $r - l$ есть по два варианта, причём $r + l$ и $r - l$ должны быть одной чётности. Для того, чтобы определить подходящий вариант, нужно проверить эту чётность. Для этого надо сделать ещё один запрос $(? \ m - 1 \ m + k)$. Если вернуло 1, то выполнено неравенство:

$$k + 1 > ((m + k) - r) + 1 + ((m - 1) - l) \Leftrightarrow r + l > 2m - 1 \Leftrightarrow \\ \Leftrightarrow r + l = 2m \wedge r - l = 2k \Rightarrow l = m - k, r = m + k.$$

Иначе получается следующее неравенство:

$$k + 1 \leq ((m + k) - r) + 1 + ((m - 1) - l) \Leftrightarrow r + l \leq 2m - 1 \Leftrightarrow \\ \Leftrightarrow r + l = 2m - 1 \wedge r - l = 2k - 1 \Rightarrow l = m - k, r = m + k - 1.$$

Последний запрос делать не надо, если $m + k > n$. В этом случае сразу получается второй вариант (с нечётными $l = m - k$ и $r = m + k - 1$).

Таким образом, алгоритм имеет следующий вид:

1. С помощью бинарного поиска находим максимальное m такое, что $(? \ 1 \ m) = 0$. (Не более 20 запросов)
2. С помощью бинарного поиска находим максимальное k такое, что $(? \ m - \lceil \frac{k}{2} \rceil \ m + \lfloor \frac{k}{2} \rfloor) = 0$. (Не более 20 запросов)
3. Если $k = 1$, то нужный отрезок — $(m - 1, m + 1)$.
4. Если $m + k > n$ или $(? \ m - 1 \ m + k) = 0$, то нужный отрезок — $(m - k, m + k - 1)$. (Не более 1 запроса)
5. Иначе нужный отрезок — $(m - k, m + k)$.

Итого не более 41 запроса.

Задача F. Алхимическое чудо

Ответ в задаче никогда не превосходит трех, это следует из проблемы Гольдбаха (https://ru.wikipedia.org/wiki/Проблема_Гольдбаха). Есть две формулировки этой проблемы:

- **Тернарная проблема Гольдбаха** — любое нечётное число, начиная с 7, можно представить в виде суммы трёх простых чисел. В 2013 году она доказана перуанским математиком Харальдом Гельфготтом;
- и более общая **Бинарная проблема Гольдбаха** — любое чётное число, начиная с 4, можно представить в виде суммы двух простых чисел. Является открытой математической проблемой, включена в список проблем Гильберта под номером 8. Но нас будет интересовать, что для всех чётных чисел, не превышающих 4×10^{18} , она выполняется, что было проверено еще в 2012 году с использованием компьютерных вычислений.

Из этого легко получить решение задачи. Если на вход поступило простое число, то ответ — 1. Если нам дали четное число или, особый случай, число $n - 2$ простое, то ответ будет равен 2. Иначе мы имеем нечетное число, для которого из тернарной проблемы Гольдбаха можно найти разложение на три простых числа, значит, ответ — 3.

Задача Г. Подарок на юбилей

Рассмотрим аналогичную задачу, но на булевом массиве (состоящем из 0 и 1). Операция Марка на этом массиве изменяет его, если только a_i и a_{i+1} оба равны 1. В таком случае $c = 1$ и после $a_i \oplus c = 0$, $a_{i+1} \oplus c = 0$. Если же любое из чисел (или оба) равны 0, то $c = 0$ и применение XOR ничего не дает. Таким образом, применение операции Марка к двум единицам превращает их в нули, при этом сумма элементов массива уменьшается на два. Кроме того, заметим, что ни при каких элементах a_i и a_{i+1} единица не может поменяться местами с нулем.

Теперь научимся находить минимальную сумму, которая и будет индексом Тани сводимости массива. Пользуясь тем, что единица не может поменяться местами с нулем, получим, что весь булев массив разбивается на несколько подмассивов из всех единиц, т.к. применение операции Марка к последнему элементу или к элементу перед первым бесполезно — это будут пары 1, 0 и 0, 1 соответственно, которые от нее не изменяются.

Упростили задачу — необходимо минимизировать сумму элементов на массиве из всех единиц. Очевидно, что вычитая по двойке из суммы элементов, мы получим в качестве итоговой суммы остаток от деления длины массива из единиц на 2, и это будет минимальной суммой (а т.е. индексом Тани) для такого массива. Найдем правильную последовательность применений. Посмотрим на массив $e = \{1, 1, 1, 1\}$. Если применить операцию Марка к $i = 2$, то получим итоговую сумму 2, но, очевидно, что можно получить сумму 0, применив операцию (в любом порядке) к $i = 1$ и $i = 3$. Сделаем вывод, что лучше применять операцию, например, последовательно к парам единиц с начала к концу массива. Это сведет массив либо к нулевому, либо массиву из нулей с единственной 1 на последней позиции.

Эта идея переносится без изменений на задачу с произвольным булевым массивом. Если применять операцию последовательно к каждому индексу от первого до предпоследнего, то для каждого непрерывного подмассива из единиц мы выполним ровно то сведение, что было описано ранее. Операцию можно применять вне зависимости от того, находимся ли мы на паре единиц или нет. Таким образом, для булевого массива можно выполнить операцию Марка, просуммировать оставшиеся значения и получить значение его индекса сводимости.

Обобщим решение до решения исходной задачи. Для этого лишь необходимо заметить, что все k -е биты исходного массива задают булев массив, для которого мы умеем искать ответ. Нетрудно понять, что применяя операцию Марка последовательно ко всем возможным парам от начала до конца, мы получим массив с минимальной суммой, а значит индекс Тани легко находится суммированием элементов полученного массива.

Сложность решения: $O(n)$.

Задача Н. Из пушки по комарам

Заметим, что при фиксированной координате Y угол монотонно зависит от координаты X . Докажем это. Найдём такую высоту $Y_{\text{swap}} > 0$, что на ней при запуске под углами α и β ($\alpha < \beta$) первый луч (R_α) пройдет не левее второго (R_β). Для начала докажем, что если до $\alpha_{\text{old}} < \beta_{\text{old}}$, то и после преломления $\alpha_{\text{new}} < \beta_{\text{new}}$. Действительно, синус — строго монотонно возрастающая функция на $[0, \frac{\pi}{2}]$, а значит

$$\begin{aligned} \alpha_{\text{old}} < \beta_{\text{old}} &\Leftrightarrow \sin(\alpha_{\text{old}}) < \sin(\beta_{\text{old}}) \Leftrightarrow \frac{\sin(\alpha_{\text{old}}) \cdot v_{i+1}}{v_i} < \frac{\sin(\beta_{\text{old}}) \cdot v_{i+1}}{v_i} \Leftrightarrow \\ &\Leftrightarrow \sin(\alpha_{\text{new}}) < \sin(\beta_{\text{new}}) \Leftrightarrow \alpha_{\text{new}} < \beta_{\text{new}} \end{aligned}$$

Теперь рассмотрим поведение лучей в конкретном слое. Во всех слоях, кроме первого, стартовая точка у R_α левее, чем у R_β (в первом они совпадают). А так как $\alpha < \beta$, у R_β происходит большее смещение по горизонтальной оси, чем у R_α , а значит разница горизонтальных координат лучей на каждой следующей границе сред будет строго увеличиваться. В начале и конце слоя луч R_α левее луча R_β , а значит, чтобы в какой-то точке внутри слоя второй луч R_α был правее, надо, чтобы два отрезка пересеклись хотя бы в двух точках, чего не может быть. Значит, такой высоты Y_{swap} не существует, а значит при увеличении угла Координата X будет строго монотонно увеличиваться.

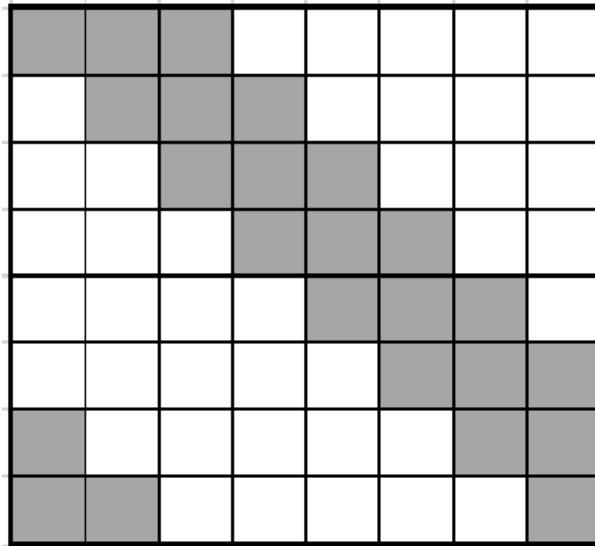
Теперь достаточно сделать бинарный поиск по ответу для угла. Имея фиксированный угол, можно просимулировать полёт луча, чтобы найти для высоты Y через точку с какой горизонтальной

координатой пройдёт луч. Для этого нужно хранить текущие горизонтальную координату X_{cur} и угол α_{cur} . Будем итерироваться по номеру слоя. При рассмотрении конкретного слоя с номером i известна его нижняя и верхняя граница (h_i и h_{i+1}) соответственно. (Предварительно укажем, что $h_0 = 0$ и $h_{n+1} = \infty$). По координате Y можно понять, находится ли точка в рассматриваемом слое, и если да, то $Y_{\text{next}} = Y$, иначе $Y_{\text{next}} = h_{n+1}$. Новую горизонтальную координату можно найти по формуле $X_{\text{new}} = X_{\text{cur}} + \tan(\alpha) \cdot (Y_{\text{next}} - h_i)$. Если надо пройти ещё один стык сред, то новый угол считается по формуле $\alpha_{\text{new}} = \arcsin\left(\frac{\sin(\alpha_{\text{cur}}) \cdot v_{i+1}}{v_i}\right)$. Дополнительно стоит сделать проверку того, что модуль аргумента арксинуса не превосходит 1. Он может превосходить, если произошло полное внутреннее отражение, а значит, надо прекращать итерацию и говорить, что луч ушёл в бесконечность по горизонтальной оси, так и не достиг нужной высоты.

После нахождения угла необходимо ещё раз запустить симуляцию и найти время полёта луча. Для этого на каждой итерации нужно прибавить к ответу $\frac{\sqrt{(x_{\text{new}} - x_{\text{cur}})^2 + (Y_{\text{next}} - h_i)}}{v_i}$.

Задача I. Нашествие жуков

Для решения данной задачи достаточно было придумать один из множества вариантов расстановки. Решение жюри расставляло стражников следующим образом при $n = 8$, $k = 3$, серыми клетками помечены точки размещения стражников:



Можно заметить, что стражники образуют лесенку и занимают позиции в i -й строке с i по $i + k - 1$, и в случае, когда они выходят за границу сетки, то счёт переходит от начала координат.

Задача J. Треугольный луч

Для удобства будем указывать в треугольных скобках координаты точки в треугольной системе, а в квадратных скобках — коэффициенты в декартовой.

Рассмотрим тройку чисел $\langle x_0, y_0, z_0 \rangle$, задающих точку. Заметим, что эту же точку можно задать, используя только две координаты $\langle x_0 - z_0, y_0 + z_0, 0 \rangle$. Это следует из того, что в треугольной системе координат $z = y - x$. Преобразуем таким образом обе точки из условия и будем работать только с парами $\langle x, y \rangle$. После вычитания начальной точки из промежуточной получим вектор, параллельный исходному лучу. Будем искать перпендикулярный ему вектор, чтобы затем отложить его от стартовой точки.

Переведём точку $\langle a, b \rangle$ из треугольных координат в декартовы. Получится точка $\left[a + \frac{b}{2}, \frac{b\sqrt{3}}{2}\right]$.

Два вектора являются перпендикулярными, если их скалярное произведение равняется нулю. В декартовых координатах скалярное произведение двух векторов $[x_1, y_1]$ и $[x_2, y_2]$ находится по формуле $x_1 \cdot x_2 + y_1 \cdot y_2$. Пусть $\langle a, b \rangle = \left[a + \frac{b}{2}, \frac{b\sqrt{3}}{2}\right]$ — исходный вектор, а $\langle c, d \rangle = \left[c + \frac{d}{2}, \frac{d\sqrt{3}}{2}\right]$ — искомый перпендикулярный вектор. Тогда они перпендикулярны, если

$$0 = \left(a + \frac{b}{2}\right) \left(c + \frac{d}{2}\right) + \frac{b\sqrt{3}}{2} \cdot \frac{d\sqrt{3}}{2} = ac + \frac{ad}{2} + \frac{bc}{2} + bd \Leftrightarrow \Leftrightarrow c(2a+b) + d(2b+a) \Leftrightarrow c : d = -(2b+a) : (2a+b).$$

Таким образом, $c = -(2b+a)$, $d = (2a+b)$ подходят. В условии просили, чтобы точка из ответа была ближайшей из возможных, поэтому разделим c и d на их наибольший общий делитель $g = \gcd(c, d)$. Тогда если стартовая вершина — это $\langle x, y, z \rangle$, то ответом будет являться точка $\left\langle x + \frac{c}{g}, y + \frac{d}{g}, z \right\rangle$.

Задача К. Заливка

Так как все цвета различные, а граф полный, то каждым перекрашиванием мы уменьшаем количество цветов на 1. Было n , нужно оставить 1, значит, сделаем $n - 1$ перекрашиваний.