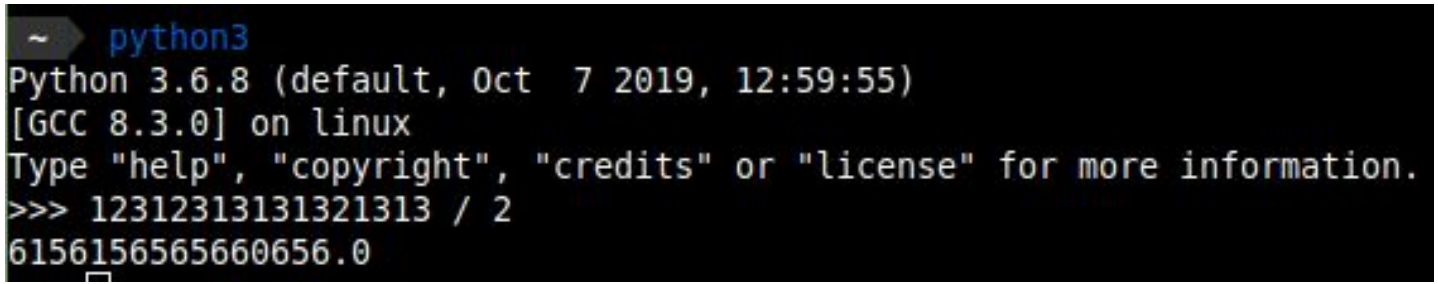
Московская городская командная олимпиада 2019 года
Лига А

Разбор задач

Задача «Целые точки»



Автор задачи: Владимир Романов
Разработчик: Григорий Резников

Формальная постановка

- Дано n прямых вида $y = x + c$ и $y = -x + c$
- Определить количество пар прямых, которые пересекаются в точке с целыми координатами

Решение

- Две прямые вида $y = x + a$ и $y = x + b$, очевидно, не пересекаются, так как их угловые коэффициенты совпадают
- Аналогично с прямыми вида $y = -x + a$ и $y = -x + b$
- Рассмотрим пару прямых $y = x + a$ и $y = -x + b$
- Они пересекаются в точке $((a - b) / 2, (a + b) / 2)$
- Отсюда видно, что точка пересечения целочисленная тогда и только тогда, когда a и b одной чётности

Решение

- Посчитаем количество прямых каждой чётности свободного члена по каждому направлению
- Тогда ответ представляется как сумма произведений чётных количеств и нечётных количеств

Обрати внимание

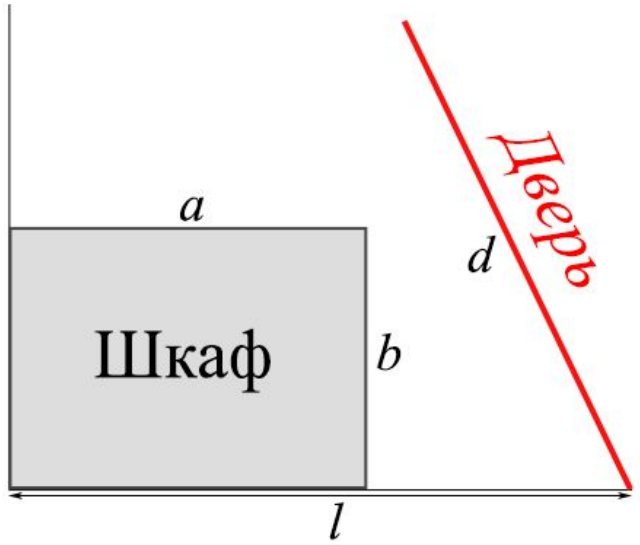
- ~~Сделано в Германии~~
- Ответ на задачу не помещается в 32-битном типе данных, поэтому требуется использовать 64-битные типы данных (long long, int64_t и другие).

Задача «Дорогой шкаф»

Автор задачи: Денис Кириенко
Разработчик: Грибов Филипп

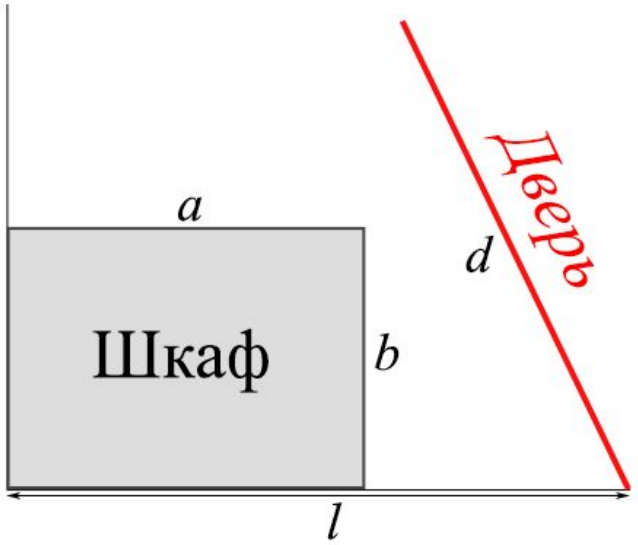
Формальная постановка

- В углу комнаты стоит шкаф размера a * b.
- На расстоянии l от угла стоит дверь длиной d
- Требуется узнать, может ли дверь при открытии задеть шкаф, или же она упрётся в стену



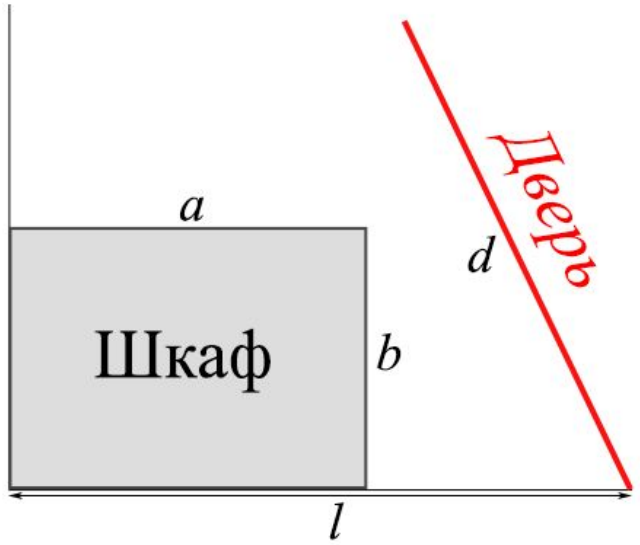
Решение

- Если дверь не заденет шкаф, то она может упереться в ближнюю или дальнюю стену
- Чтобы дверь упёрлась в ближнюю стену, сумма длин двери и шкафа должна быть меньше расстояния от шкафа до стены, т.е. a + d < l



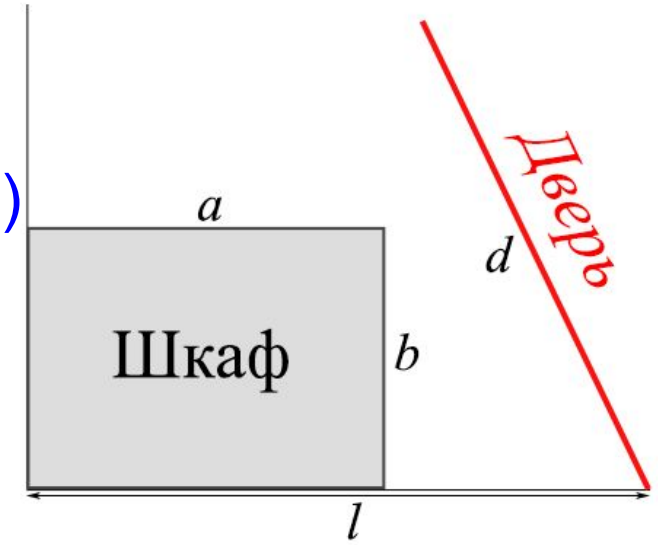
Решение

- Чтобы дверь упёрлась в дальнюю стену, надо, чтобы отрезок, выходящий из основания двери, проходящий через угол шкафа и заканчивающиеся на другой стене, имел длину меньше d



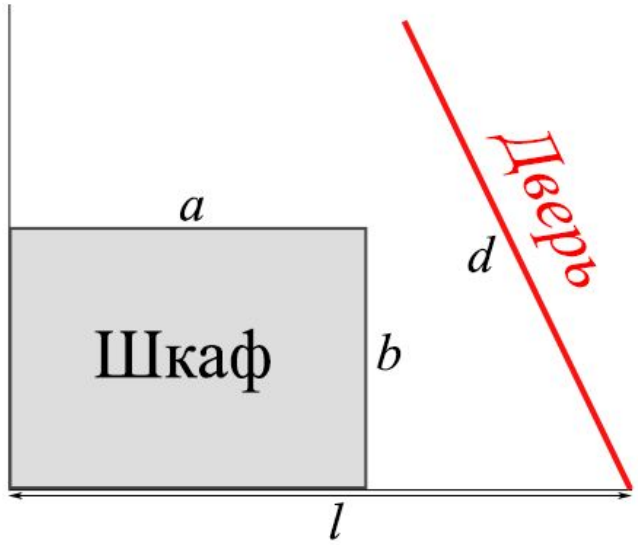
Решение

- По теореме Пифагора, длина отрезка от основания двери до угла шкафа равна sqrt((l - a)² + b²)
- Тогда длина этого отрезка, продлённого до стены, равна sqrt((l - a)² + b²) / (l - a) * l.
- Нам надо сравнить это с d



Решение

- Чтобы не было проблем с точностью, возведём обе части в квадрат и перенесём деление в другую сторону. Получим что надо сравнить $((l - a)^2 + b^2) * l^2$ и $d^2 * (l - a)^2$



Задача «Дороги в стране»

Автор и разработчик задачи: Вячеслав Наймушин

Формальная постановка

- Задан взвешенный ориентированный граф
- Для каждой вершины требуется найти путь из неё в вершину 1, в котором минимальный вес ребра максимален

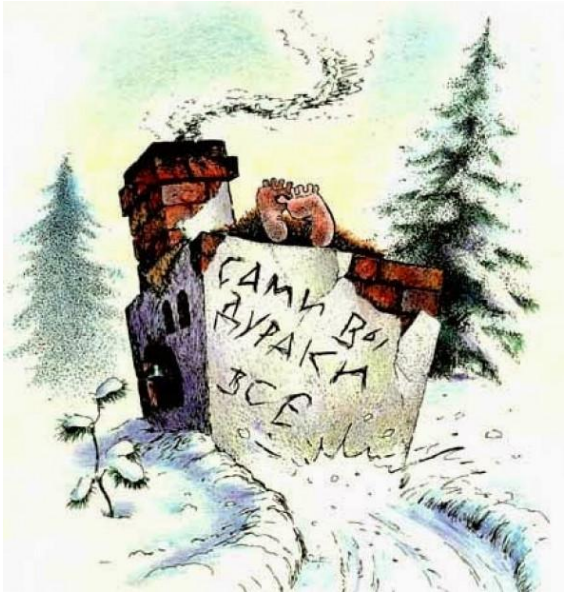
Решение

- Будем добавлять рёбра в порядке убывания веса и поддерживать ответы для всех вершин
- Изначально ответ для всех вершин не определён.
- Если добавленное ребро из A в B веса X, и для вершины B посчитан ответ, а для вершины A ответ не посчитан, то ответ для вершины A равен X

Решение

- Как только мы посчитали ответ для новой вершины, необходимо посмотреть на все ребра ведущие в нее (т.е. обратные ребра). Если ребро начинается из вершины, для которой ответ еще не был посчитан, то присваиваем ответ X и тоже смотрим на ребра ведущие в нее
- Таким образом каждое ребро “посчитано” дважды, а значит итоговая сложность O(m log m) на сортировку ребер и O(m) на обходы вершин

Задача «Иванушка-дурачок и теория вероятностей»



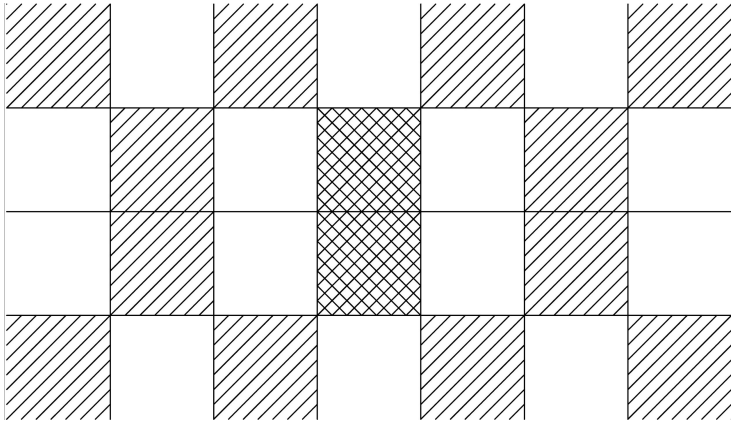
Автор и разработчик задачи: Владимир Романов

Формальная постановка

- Есть клетчатое поле n на m
- Нужно подсчитать число черно-белых раскрасок этого поля таких, что для любой клетки верно, что среди ее соседей по стороне не более одной клетки такого же цвета, что и она

Ключевая идея

- Пусть есть вертикальная одноцветная пара клеток



Тогда их соседи должны быть другого цвета. Аналогично соседи соседей и т. д.

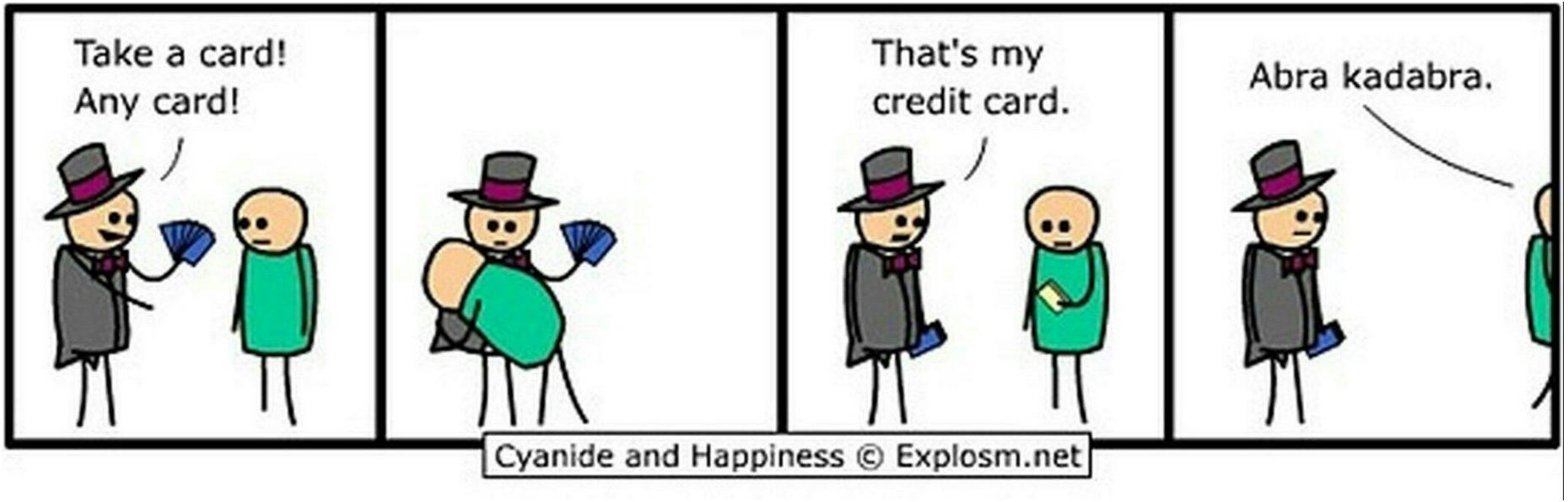
Таким образом некоторые соседние строчки разбиваются на пары

Число способов разбить некоторые соседние строки на пары равно F_n

Решение

- Есть 2 варианта цвета угловой клетки
- Число способов разбить соседние строки на пары F_n
- Число способов разбить соседние столбцы на пары F_m
- Шахматная раскраска учитывается дважды
- Итоговый ответ $2(F_n + F_m - 1)$

Задача «Фокус с делением и умножением»



Автор и разработчик задачи: Иван Сафонов

Формальная постановка

- Есть массив различных целых чисел
- За одну операцию можно разделить нацело или умножить один элемент массива на целое число
- На каждое число нельзя умножать/делить больше одного раза
- За какое минимальное число операций можно сделать все числа равными?

Ключевая идея

- За n действий всегда можно сделать все числа равными
- Меньше чем за n - 1 нельзя
- Если хотим за n - 1 действие, то все числа надо сделать равными какому-то числу из массива
- Надо найти числа массива, такие что все остальные либо делятся на него, либо его делитель и проверять, можно ли все остальные свести к нему

Решение

- Рассмотрим позиции i, такие что gcd(a[i+1], …, a[n]) делится на a[i]. Таких позиций не больше чем log(A).
- Каждую позицию можем проверить за O(n log n) или O(n), проверив, что числа (a[i] / a[1]), …, (a[i] / a[i-1]), (a[i+1] / a[i]), …, (a[n] / a[i]) целые и различные.
- Хотя бы одна позиция подошла - ответ n - 1, иначе n
- Полученное решение работает за *O(n log A log n) или O(n log A)*

Задача «Личность широких взглядов»

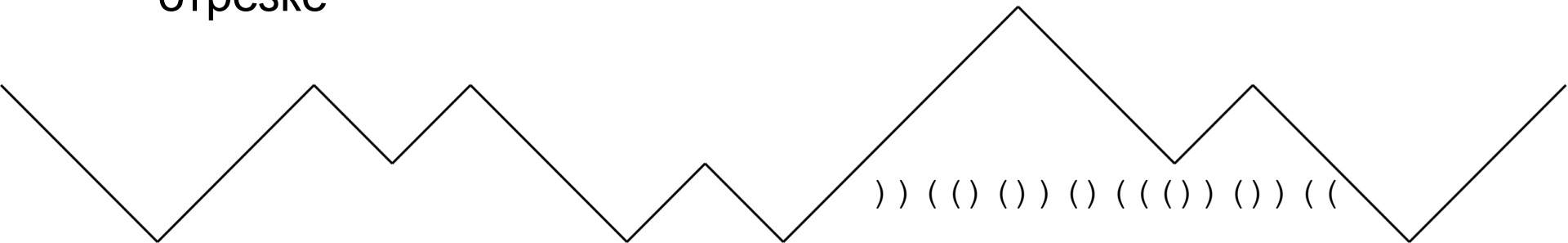
Автор задачи: Григорий Резников
Разработчик: Максим Деб Натх

Формальная постановка

- Дана скобочная последовательность
- Необходимо отвечать на два запроса:
  - Сменить i-ю скобку на парную ей
  - Ответить на вопрос о количестве циклических сдвигов отрезка [l;r], являющихся правильными скобочными последовательностями (ПСП)

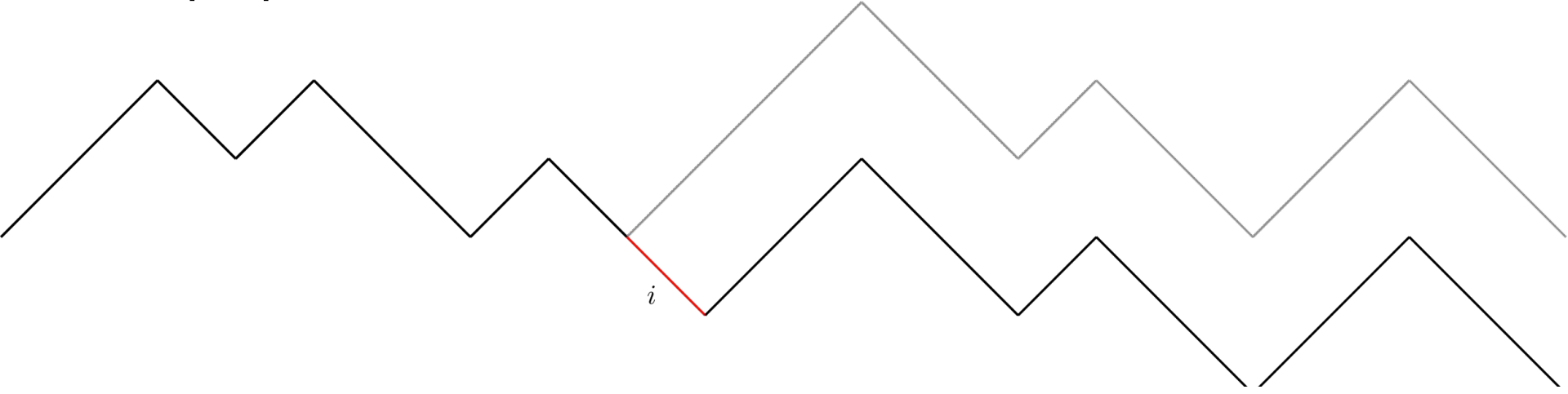
Общие замечания

- Если число открывающих скобок не равно числу закрывающих, ответ нулевой
- Префиксный баланс — разность числа открывающих и закрывающих скобок на префиксе
- Заметим, что ответ на вопрос о количестве циклических сдвигов-ПСП есть число минимальных балансов на отрезке



Общие замечания

- Операция смены i-й скобки `(` на `)` уменьшает все префиксные балансы с i+1-го на 2
- Операция смены i-й скобки `)` на `(` увеличивает все префиксные балансы с i+1-го на 2



Решение

- Будем поддерживать структуру данных, позволяющую делать прибавления на отрезке и находить минимум на отрезке (например, дерево отрезков)
- На операцию смены i+1-й скобки будем прибавлять или вычитать 2 на отрезке балансов с i-го по последний
- На запрос [l, r], убедимся, что l-й баланс равен r+1-му, если это так, вернём количество минимумов на этом отрезке, иначе — ноль
- Решение работает за $O(n + q \log n)$

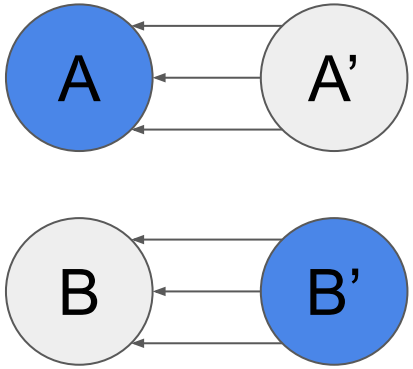
Задача «Конкурс котиков»

Автор задачи: Василий Алфёров
Разработчик: Николай Будин

Формальная постановка

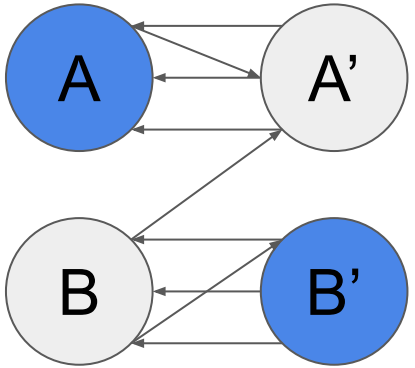
- Дан двудольный граф
- В обеих долях по n вершин
- Дано полное паросочетание
- Нужно найти независимое множество размера n, не равное целиком ни одной из долей

Решение



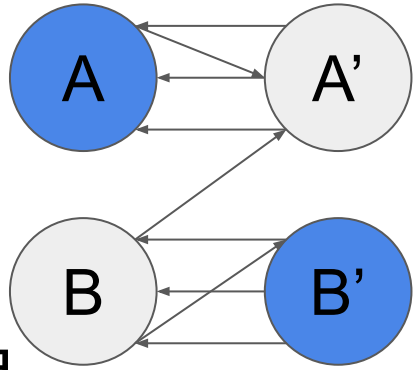
- A — множество вершин независимого множества из левой доли
- A’ — множество вершин, связанных с A ребрами из паросочетания
- Независимое множество должно также содержать все вершины из B’
- Ориентируем ребра из паросочетания справа налево, а остальные ребра — слева направо

Решение



- Ребра из паросочетания ведут из A’ в A и из B’ в B
- Ребра не из паросочетания могут вести из A в A’, из B в B’, из B в A’
- Ребер из A в B’ быть не может, потому иначе множество A ∪ B’ не будет независимым
- Можно заметить, что из A не достижимо B’

Решение

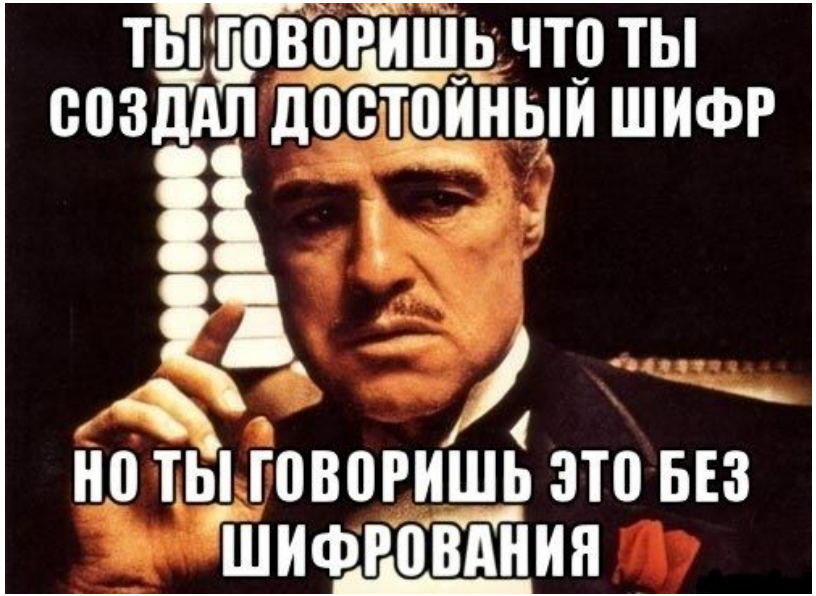


- Найдем компоненты сильной связности в построенном ориентированном графе
- Если компонента связности одна, решения не существует, потому что из вершин A не могут быть достижимы вершины B’, значит одно из этих множеств пусто
- Если компонент связности хотя бы две, можно выбрать в качестве B’ все вершины правой доли, находящиеся в любой компоненте, являющейся истоком

Решение

- Исключение: $n = 1$
- Две компоненты сильной связности, но ответа нет

Задача «XOR шифрование»



Автор и разработчик задачи: Иван Сафонов

Формальная постановка

- Есть множество A целых чисел от 0 до $2^{20} - 1$ и некоторое целое число k
- Мы хотим уметь удалять добавлять числа в множество
- После каждой операции надо сообщать минимальное mex(A xor x) по всем $0 \leqslant x \leqslant k$

Ключевая идея

- Свести задачу для $0 \leqslant x \leqslant k$ к небольшому числу задач, таких что $(k + 1)$ это степень 2
- Решить задачу в этом случае

Решение

- Если $x < k + 1$, то все такие x описываются как фиксированный префикс двоичной записи $k + 1$, после идет 0, а дальше какие угодно биты
- Получаем $\log(k)$ префиксов, остальные биты выбираются любыми
- Тогда будем хранить множества $(A \text{ xor } p)$ для каждого такого префикса p
- Получим, что мы свели задачу к $\log(k)$ задачам, в которых мы можем выбирать $0 \leqslant x < 2^{20}$

Решение

- Если мы ограничены некоторой степенью двойки 2^t, то заметим, что ответ равен минимальному s, такому что среди чисел отрезка $[s*2^t, (s+1)*2^t-1]$ нету хотя бы одного числа из множества.
- Поддерживаем сет таких s, что их отрезок не полностью заполнен, так сможем находить минимальное.
- Полученное решение работает за время *O(n log k log n)*

Альтернативное решение

- Можно заметить для для каждого числа y из множества !A нас интересует такое число $0 \leqslant x \leqslant k$ что x xor y минимален
- Наш ответ - минимум по всем таким x xor y
- Эти числа можно преподсчитать в самом начале за *O(n log n)* и сложить в отсортированное мультимножество

Альтернативное решение

- Дальше на каждый запрос нам надо либо удалять число из множества, либо добавлять число обратно в множество
- Полученное решение работает за время $O(n \log n)$

Задача «Очередь в поезде»

Автор задачи: Михаил Пядёркин
Разработчик: Евгений Шульгин

Формальная постановка

- Есть очередь из людей, где каждый человек обрабатывается за p минут
- Человек i захочет попасть в очередь на t_i-й минуте.
- Он сделает это в первый момент $\geqslant t_i$, как только в очереди не будет людей с номерами меньше i
- В случае, если одновременно могут выйти несколько людей, выходит имеющий наименьший номер
- Для каждого человека нужно сказать, когда он будет обработан

Идея

- Задачу легче всего решать через поддержку *событий*
- *Событие* - тройка чисел *(time, type, index)*
- *time* - время события.
- *type* - тип события (пусть *0* - человек вернулся из очереди, *1* - человек захотел в очередь).
- *index* - номер человека.
- Для корректности события нужно обрабатывать в отсортированном порядке.

Решение (начало)

- Заводим несколько переменных. “Множество” - std::set или std::priority_queue.
- *events* - множество событий.
- *want* - множество людей, хотящих в очередь.
- *in_queue* - множество людей, стоящих в очереди.
- *queue_time* - время, когда очередь опустеет (чтобы можно было определить время выхода для нового человека).
- *cur_time* - текущее время (последнего обработанного события).

Решение (продолжение)

- Вначале есть *n* событий *(t_i, 1, i)*, кладем их в *events*.
- Пока *events* не пусто, берем первое событие и обрабатываем его.
- Если тип *1*, кладем *i* в *want*.
- Если тип *0*, удаляем *i* из *in_queue*.

Решение (конец)

- После обработки каждого события проверяем, можно ли кого-то добавить в очередь.
- Пусть *x* - наименьший элемент *want*.
- Если *in_queue* пустой или *x* меньше наименьшего элемента *in_queue*, то добавляем x в in_queue и событие *(max(cur_time, queue_time) + p, 0, x)* в *events*.
- Решение работает за *O(n log n)*.

Задача «Марио и мировой рекорд»

Автор задачи: Владимир Романов
Разработчик: Роман Горбунов

Формальная постановка

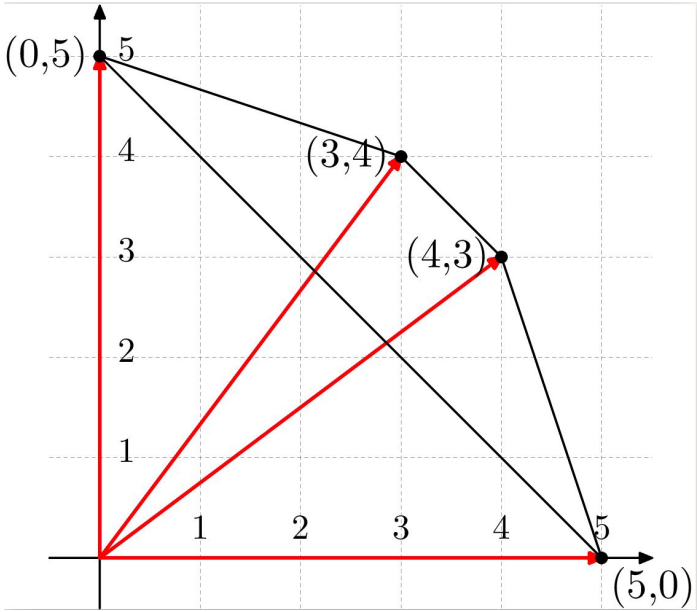
- Даны n квадратов длин отрезков
- Требуется построить ломаную из этих отрезков, вершины которых будут в целочисленных координатах
- Расстояние от точки (0, 0) до (x_n, y_n) - максимально

Идеи

- Так как квадрат длин отрезков $a_i \le 10^6$, то длина по координатам x и $y \le 1000$ ($x^2+y^2 \le 10^6$)
- Для всех длин *len* до 10^6 можно предподсчитать набор векторов (x, y), имеющих длину *len*
- Нас интересуют варианты, находящиеся в положительной четверти

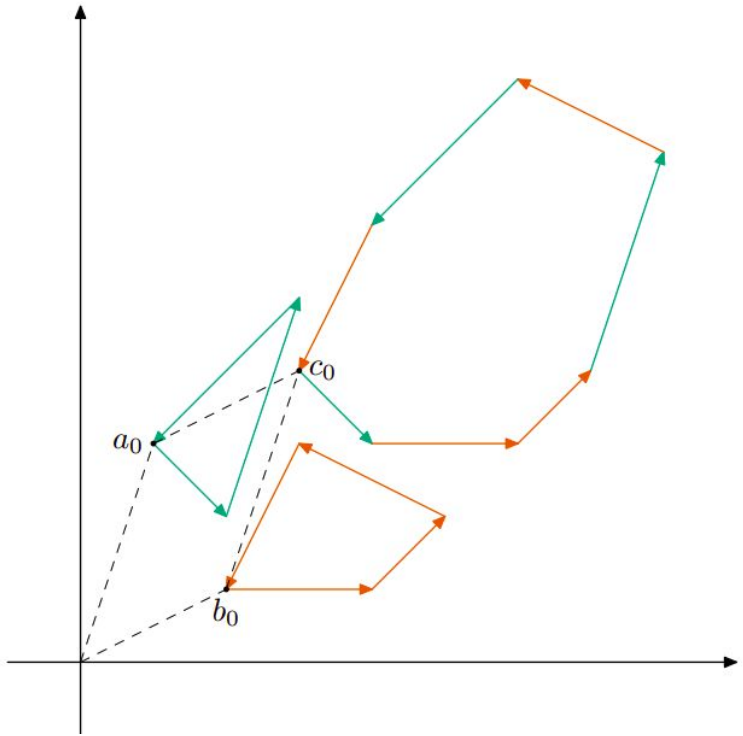
Решение (этап 1)

- Для каждого отрезка подсчитать выпуклую оболочку его возможных представлений в четверти.



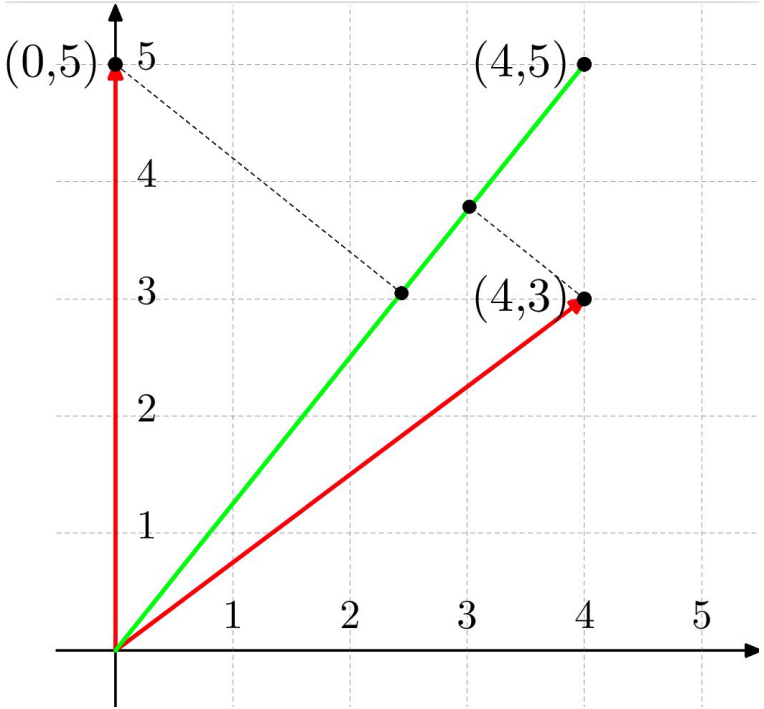
Решение (этап 2)

- Все точки, которые мы можем получить содержатся в многоугольнике суммы Минковского выпуклых оболочек
- Самая дальняя точка этого многоугольника - ответ



Решение (этап 3)

- Для каждой длины отрезка вектор с максимальным скалярным произведением входит в ответ.



Оценка сложности

Этап 1) $C \cdot 1000^2 = C \cdot 10^6$ для построения всех возможных векторов

Обозначим за sz сумму размеров выпуклых оболочек для длин отрезков

Этап 2) $O(sz \cdot log(sz))$ для построения суммы Минковского

Этап 3) $O(sz)$ для построения ломанной по направлению

Так как максимальный размер у выпуклой оболочки оказался 32, $sz = 32 \cdot n \leq 3.2 \cdot 10^6$

Результирующая сложность $C \cdot 10^6 + O(sz \cdot log(sz))$

Задача «Черепашка» Div. A

Автор задачи: Владимир Романов
Разработчик: Дмитрий Саютин

Постановка задачи

- Расставить заданное число нулей и единиц в матрице 2 x n
- Минимизировать максимальный путь черепашки, которая может ходить только вправо и вниз

Ключевая идея

Start		100	10		
					Finish

Start		10	100		
					Finish

В результате замены ни один путь не увеличился.

Некоторые уменьшились

Ключевая идея

Значит можно отсортировать и верхнюю и нижнюю строку таблицы:

10	20	30	40	50	60
65	55	45	35	25	15

Но как выглядит итоговый путь? И как распределить числа на две строки?

С каких пор это стало лучше чем это?

10	20	30	40	50	60
65	55	45	35	25	15

10	20	30	40	50	60
65	55	45	35	25	15

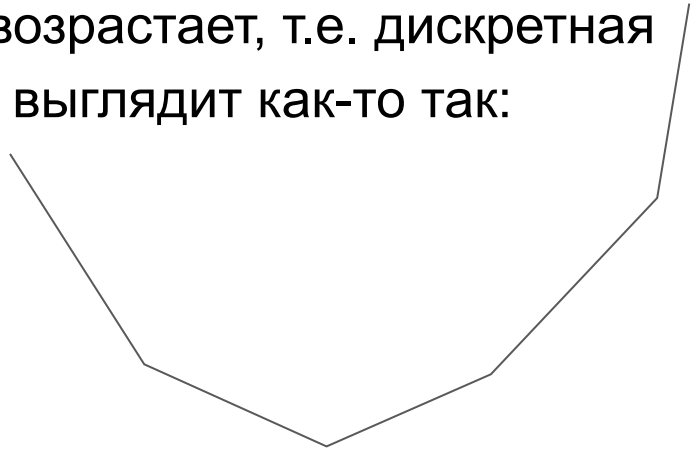
10	20	30	40	50	60
65	55	45	35	25	15

Формализм

Пусть первая строка - это $x_1, x_2, \ldots, x_n$, а вторая - $y_1, y_2, \ldots, y_n$.

Тогда цена пути $i + 1$ минус цена пути i равна $x_{i+1} - y_i$.

x_i возрастает при $i{+}{+}$, y_i убывает, $x_{i+1} - y_i$ возрастает, т.е. дискретная производная возрастает, такая функция выглядит как-то так:



=> черепашка выберет

или первый путь, или последний

Решение

- Итого, если выкинуть из рассмотрения угловые элементы (в которые оптимально поместить минимум), то надо разбить n - 2 целых числа на две группы поровну, минимизируя максимальную сумму в группе
- Рюкзак. Состояние: <количество предметов в группе, вес группы>
- Итого $n * sum$ состояний, и $n^2 sum = n^3 maxa$ переходов

Решение

- Уже ОК, но можно ускорить, применив битовое сжатие
- Тогда n^3 maxa / 64

Задача «Жулик, не воруй»

Автор и разработчик задачи: Владимир Романов

Формальная постановка

- Дано связный граф на n вершинах и m ребрах, в котором нет кратных ребер и петель
- Требуется найти непустой и несовпадающий набор вершин такой, что если оставить данные вершины и ребра между ними, для каждой оставленной вершины верно, что ее степень mod 3 совпадает с изначальной mod 3

Обозначения и случай 1

- Пусть вершина-x — вершина степень которой mod 3 равна x
- Если в графе есть вершина-0, то ее можно выбрать
- Исключением является граф из одной вершины

Случай 2

- На данный момент мы уже считаем, что в графе нет вершин-0
- Если есть цикл на вершинах-2, то существует несократимый цикл на вершинах-2
- В этом случае ответом является этот несократимый цикл
- Исключением является граф, который является циклом

Случай 3

- На данный момент мы уже считаем, что в графе нет вершин-0 и циклов на вершинах-2
- Если есть две вершины-1, то между ними есть путь
- Тогда существует несократимые путь на вершинах-2, заканчивающийся на вершинах-1
- В этом случае ответом является этот несократимый путь
- Исключением является граф, который является бамбуком

Случай 4

- На данный момент мы уже считаем, что в графе нет вершин-0, циклов на вершинах-2 и не более одной вершины-1
- Ноль вершин-1 не может быть
- Вершины-2 образуют лес

Случай 4

- Предположим лес представляет из себя ровно одно дерево
- k — кол-во вершин-2
- Их суммарная степень mod 3 равна 2k mod 3
- Степень вершины-1 равна суммарной степени вершин-2 минус удвоенное число ребер между вершинами-2
- Степень вершины-1 mod 3 равна 2k - 2(k-1) = 2
- Противоречие

Случай 4

- Лес на вершинах-2 содержит хотя бы 2 дерева
- В этом случае ответом является вершина-1 и два пути на вершинах-2
- Исключением является граф, который является объединением двух циклов

Конец

- В итоге мы разобрали все случаи, и каждый из случаев решается за линейное время при помощи dfs или bfs
- Асимптотика решения: $O(m)$