Председатель жюри: Овчаров Иван
Состав жюри: Туров Макар
Мизёв Андрей
Девятериков Иван
Поздняков Никита

Спонсоры олимпиады: PARMA Technologies Group
Knoema
Белоногов Иван

Благодарность тем, кто протестировал задачи: Цаплин Александр
Халикшин Тимур
Лизунов Михаил
Базилевич Григорий

Отдельная благодарность: Шульгина Галина Михайловна

Задача A. Тяжелые будни полиции

Автор: Кориненко Матвей
Разработчик: Мизёв Андрей

Давайте поделим n монет пополам и на одной из половин запустим машину. Если фальшивая среди тех, с которыми мы ее запускали — продолжим работу с этой половиной, иначе с той, которую мы не помещали в машину. Повторим это действие до того момента, пока в одной из кучек не будет 1 монеты. За все это время мы совершим $\lceil \log_2 n \rceil$ запусков машины. Это и есть ответ. Обратите внимание, вычисление значения двоичного логарифма средствами арифметики с плавающей точкой может быть недостаточно точным!

Задача B. Кузнечик против Монстра

Автор и разработчик: Девятериков Иван

Заметим, что минимальным прыжком, который может совершить Кузнечик, является наибольший общий делитель всех доступных прыжков. Т.е. всегда можно придумать комбинацию прыжков, которая совершит прыжок на НОД доступных прыжков. Также нетрудно доказывается, что меньше, чем на НОД прыгнуть нельзя.

Тогда имеем такую задачу: удалить число из массива или найти НОД оставшихся элементов массива. Если развернуть данную задачу (заменить удаление добавлением), то ее решение станет тривиальным: НОД массива с добавленным числом будет равен НОДу НОДа массива без числа и добавляемого числа. Таким образом мы приходим к решению. Необходимо сохранить массив, все запросы и проимитировать их с конца, сохранить ответы и вывести в нужном порядке.

Задача C. Система Нелинейных Неалгебраических Уравнений

Автор: Девятериков Иван
Разработчик: Поздняков Никита

Заметим, что мы можем представить СННУ как ориентированный невзвешенный граф (рёбра идут из большего числа в меньшее). Тогда поймём, что если в нём есть цикл то такая система не имеет решений. Иначе, заметим что нам нужно как минимум столько чисел, какова длинна самого длинного пути в графе.

Чтобы найти его, найдём топологическую сортировку графа, а затем будем идти по топологической сортировке с конца, обновляя значение максимального пути начинающегося в этой вершине, так как все её потомки уже посчитаны. Сравнив максимальное из этих значений с k узнаем ответ.

Задача D. Заполни таблицу

Автор и разработчик: Мизёв Андрей

Рассмотрим два пути: один идёт до (x, y), потом идет в $(x+1, y)$, потом в $(x+1, y+1)$, потом как-то до конца, второй так же идет до (x, y), потом идет в $(x, y+1)$, потом $(x+1, y+1)$, потом также как и первый идет от $(x+1, y+1)$ до конца. Эти два пути имеют одинаковую сумму до (x, y) включительно и от $(x+1, y+1)$ включительно, а разные слагаемые — только числа в $(x+1, y)$ и $(x, y+1)$. Тогда чтобы суммы на всех (в том числе этих двух) путях были одинаковы число в $(x+1, y)$ должно быть равно числу в $(x, y+1)$, причем это верно для всех существующих $(x+1, y)$ и $(x, y+1)$. Это значит, что во всех диагоналях, перпендикулярных главной (идущих «вправо-вниз»), должны быть равные значения. Заметим, что этого условия достаточно, т.к. любой путь проходит через каждую диагональ ровно один раз, поэтому сумма в любом пути будет равна сумме всех значений в различных диагоналях.

Значит, для нахождения ответа необходимо узнать, можно ли так дописать числа, чтобы во всех диагонялях, перпендикулярных главной, были равные значения. Если существуют 2 заполненные клетки из одной диагонали с разными значениями, то это, очевидно, это невозможно. Если таких клеток не существует, то заполнение возможно. Проверить наличие такой пары клетом можно отсортировав массив заданных клеток по возрастанию диагонали (заметим, что диагонали, перпендикулярные главной, задаются суммой координат клетки).

Задача E. Сладкие чиселки

Автор и разработчик: Туров Макар

Заметим, что не может быть такой последовательности длиннее 200 чисел, если не все из них нули. И решим за n^3.

Докажем этот факт: Так как последовательность однозначно обратима (по двум соседним элементам можно понять все предыдущие), то ни с какого момента нули начаться не могут (тогда все предыдущие тоже нули). Если с какого-то момента начинаются числа одного знака (а это случится если будут подряд 2 числа одного знака или одно будет нулем), то дальше через 50 позиций числа возрастут больше чем до 10^9. Тогда, допустим что знаки будут чередоваться хотя бы до позиции 51. Тогда:

a,b - начальные числа. d_n - n-тое число последовательности. Оно, как можно догадаться, равно $x_n a + y_n b$, где x_n, y_n - какие-то натуральные числа, зависящие только от n, но не от a и b. Взяв $a = 0; b = 1$ получаем $x_i = f_{n-2}$, где f_n — n-тое - число Фибоначчи, аналогично $y_n = f_{n-1}$. Тогда, получаем что Если $a > 0; b < 0$, то $d_{99} = f_{97} \times a + f_{98} \times b < 0$;$d_{100} = f_{98} \times a + f_{99} \times b > 0$, тогда $\frac{f_{99}}{f_{98}} \times b < a < \frac{f_{98}}{f_{97}} \times b$. Такое условие не выполняется при $a \in \mathbb{N}$. Поэтому, 100 чисел такой последовательности не будут знакопеременными. Тогда, раз знакопеременность прекратится за 100 чисел, а из ограничений они выйдут ещё через 100, то 200 чисел, из которых хотя бы одно не 0 при заданных ограничениях не будут задавать d_n, чтд..

Задача F. Подмастерье Ваня

Автор: Туров Макар
Разработчик: Овчаров Иван

Подсчитаем массив f[t][i][j], в котором будет храниться сумма максимального подотрезка в строке i, причём если t = 0, то этот подотрезок не содержит клетку (i, j), если же t = 1, то подотрезок содержит эту клетку. Чтобы подсчитать f[0][i][j] посчитаем подотрезок с максимальной суммой слева и справа от j, тогда f[0][i][j] - это максимум из этих подотрезков. Для подсчёта f[1][i][j] просуммируем префикс с максимальной суммой, начинающийся в позиции j + 1, суффикс с максимальной суммой, заканчивающийся в позиции j - 1, и a[i][j]. Аналогично подсчитаем массив s[t][i][j] - сумма максимального подотрезка в столбце j. Тогда для нахождения ответа пройдём по всем клеткам и найдём максимум по всем значениям f[0][i][j] + s[0][i][j], f[1][i][j] + s[0][i][j], f[0][i][j] + s[1][i][j], f[1][i][j] + s[1][i][j] - a[i][j], вычитаем a[i][j], так как клетка учитывается дважды на пересечении.

Задача G. Гномы трудяги

Автор: Титов Федор
Разработчик: Овчаров Иван

Для понятности будем называть **цветом вершины** номер дела, которым занимается гном в этой вершине.

Переформулируем задачу: *найти все вершины, такие, что если за них подвесить дерево, то все вершины одного цвета будут в поддереве одного ребенка корня.* Назовём вершину (или поддерево) **хорошей (хорошим)**, если все цвета, которые есть в её поддереве, встречаются только в её поддереве. В терминах первоначальной задачи, вершина хорошая, если все гномы, живущие в поддереве, не теряют доступ друг к другу, при выборе этой вершины

Переподвешивать за каждую вершину нельзя, поэтому подвесим дерево за любую вершину выдвинем условие, когда любая вершина v входит в ответ: все её дети должны быть хорошими. Заметим, что если все дети v хорошие, то и все цвета, которые встречаются в наддереве, будут встречаться только там

Как быстро узнать является ли вершина v хорошей? Давайте поддерживать $colors_v$ - map из пар *цвет-количество*, $full_v$ - сколько цветов встречаются только в поддереве v. Теперь вершина v хорошая в двух случаях:

- $colors_v.size() == full_v$
- $colors_v.size() == full_v + 1$ и, если c - цвет предка вершины v, то $colors_{v,c} + 1 == color\_count_c$. Иными словами, предок вершины v - единственная вершина цвета c не из поддерева v.

Как поддерживать $colors_v$? Воспользуемся переливайкой. Будем каждый раз, используя $.swap(colors_u)$, «перекидывать» самый большой map в $colors_v$, а все остальные добавлять обычным проходом. Если при добавлении цвета c, $colors_{v,c}$ стало равно $color\_count_c$, то $full_v++$. Таким образом, мы умеем узнавать для всех вершин, являются ли они хорошими. Теперь достаточно для каждой вершины v узнать являются ли все её дети хорошими и, если да, вывести её

Итоговая сложность: $O(n \log^2 n)$