

Задача А. Chocorev. Шоколадка-революция

Имя входного файла: `chocorev.in`
Имя выходного файла: `chocorev.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Аким получил золотую медаль на IOI, за что от правительства ему положена благодарность в виде шоколадки. На данный момент имеется N видов шоколадок, которые производит государственная шоколадная фабрика. У каждой шоколадки есть стоимость. Кроме того, у каждого вида, кроме шоколадки «Юлька», есть вид-прародитель. Известно, что прародителем может быть только шоколадка, которая была выпущена хронологически раньше. Исторически самый первый вид — «Юлька». Чиновник и Аким выбирают шоколадку Акиму в подарок. При этом цель первого — сэкономить, второго — получить настолько дорогую шоколадку, насколько это возможно. Начинается выбор с «Юльки». Аким за ход либо берет текущую шоколадку, либо меняет свой выбор на шоколадку-потомка (если это возможно). Чиновнику же кажется, что позже выпущенные виды хуже и дешевле, поэтому он меняет выбор на шоколадку-потомка (если это возможно), либо покупает Акиму текущую шоколадку (если потомков нет). Первым ходит Аким. Определите, на шоколадку какой стоимости может претендовать Аким.

Формат входного файла

Виды шоколадок занумерованы в некотором произвольном порядке, «Юлька» имеет номер 1. В первой строке входного файла находится одно число N — количество видов шоколадок ($0 < N \leq 200\,000$). Далее следуют N строк, в каждой из которых заданы два числа P_i и C_i — номер сорта-прародителя i -ого сорта и стоимость i -ого сорта ($0 \leq P_i \leq N$, $0 < C_i \leq 1\,000\,000\,000$; для «Юльки», у которой нет прародителя, указано $P_1 = 0$);

Формат выходного файла

Выведите в выходной файл одно число — стоимость шоколадки, которую получит Аким при правильных действиях как чиновника, так и своих.

Пример

chocorev.in	chocorev.out
8 0 4 5 3 1 2 5 1 1 5 4 8 3 6 3 7	6

Задача В. Комфортабельная рассадка

Имя входного файла: `comfort.in`
Имя выходного файла: `comfort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Автор задачи: Ivan Kazmenko

В парке стоит скамейка на n мест. Изначально она пуста. Две команды играют в игру, по очереди *комфортабельно* сажая на скамейку по одному человеку. Человек садится *комфортабельно*, если на соседних местах ни справа, ни слева от него непосредственно никого нет. Проигрывает та команда, которая не может *комфортабельно* посадить очередного человека на своём ходу.

Дано n — количество мест на скамейке. Какая команда выигрывает при правильной игре?

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите “FIRST”, если при правильной игре выигрывает первая команда, и “SECOND”, если вторая.

Примеры

<code>comfort.in</code>	<code>comfort.out</code>
2	FIRST
4	SECOND

Задача С. Жестокая задача

Имя входного файла: `cruel.in`
Имя выходного файла: `cruel.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Штирлиц и Мюллер стреляют по очереди. В очереди n человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

Формат входного файла

Входной файл содержит единственное число n ($2 \leq n \leq 5000$) — количество человек в очереди.

Формат выходного файла

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова **Mueller**. Иначе в первой строке необходимо вывести слово **Schtirlitz**, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

Пример

<code>cruel.in</code>	<code>cruel.out</code>
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Задача D. Игры на графе

Имя входного файла:	gg.in
Имя выходного файла:	gg.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Противник начал e2-e4. Я
проанализировал его архитектуру и сдался

Из мемуаров 20-го чемпиона мира Фрица
Рыбкина

Виталий Вальтман

Прибыв на место, Ааз тут же потребовал организовать совещание букмекеров, на котором он изложит свой план.

— Главная задача, — начал Ааз своё выступление перед букмекерами, — научиться использовать достижения прогресса. Мы планируем запуск множества новых видов соревнований, что — вполне возможно — приведёт к тому, что появятся какие-то игры по правилам, придуманным не нами. А значит, необходимо уметь быстро выяснять, насколько эти правила могут быть нам полезны.

— А можно ли хотя бы в общем пояснить, как это будет делаться? — последовал вопрос из зала.

— Вот пример задачи, решив которую, мы сможем разобраться с целым классом игр. Дан ориентированный граф некоторой игры для двух игроков и начальная позиция в ней. Напомним, что в игре на графе игрок имеет право походить из позиции в любую позицию, в которую есть ребро из текущей. Игроки ходят по очереди; проигрывает тот, кто не может сделать ход. Требуется проверить, верно ли, что при любой игре сторон всегда выигрывает первый игрок.

Формат входного файла

Во входном файле содержится описание одного или нескольких тестов. В первой строке каждого теста заданы число вершин V и число рёбер E ($1 \leq V \leq 100\,000$, $1 \leq E \leq 100\,000$), а также номер начальной позиции a ($1 \leq a \leq V$). Далее следуют E строк — описания рёбер в формате $u_i v_i$ ($1 \leq u_i, v_i \leq V$), что означает наличие ребра, направленного из вершины u_i в вершину v_i . Файл завершается тремя нулями. Сумма всех E по всем тестам не превосходит 100 000, количество тестов в файле не превосходит 1000.

Формат выходного файла

Следуйте формату примера максимально точно — проверка производится автоматически.

Пример

gg.in
3 2 1 1 2 1 3 1 1 1 1 1 4 3 1 1 2 1 3 3 4 0 0 0
gg.out
First player always wins in game 1. Players can avoid first player winning in game 2. Players can avoid first player winning in game 2.

Задача Е. Производство графов

Имя входного файла: `graphprod.in`
Имя выходного файла: `graphprod.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Автор задачи: Andrey Lopatin

Пусть дан ориентированный ациклический граф. Стандартная игра на графе заключается в следующем: изначально на одной из вершин графа (называемой начальной позицией) стоит фишка. Двое игроков по очереди двигают её по рёбрам. Проигрывает тот, кто не может сделать ход.

В теории игр часто рассматриваются более сложные игры. Например, прямое произведение двух игр на графах. Прямое произведение игр — это следующая игра: изначально на каждом графе в начальной позиции стоит по фишке. За ход игрок двигает обе фишки по рёбрам (каждую фишку двигает в собственном графе). Проигрывает тот, кто не может сделать ход. То есть тот, кто не может сделать ход хотя бы в одной игре.

Ваша задача — опеределить, кто выиграет при правильной игре.

Формат входного файла

На первой строке будут даны числа N_1 и M_1 — количество вершин и рёбер в первом графе ($1 \leq N_1, M_1 \leq 100\,000$). На следующих M_1 строках содержится по два числа x и y ($1 \leq x, y \leq N_1$).

В следующих $M_2 + 1$ строках задан второй граф в том же формате.

Заканчивается входной файл списком пар начальных вершин, для которых нужно решить задачу. На первой строке задано число T ($1 \leq T \leq 100\,000$) — количество пар начальных вершин. В следующих T строках указаны пары вершин v_1 и v_2 ($1 \leq v_1 \leq N_1, 1 \leq v_2 \leq N_2$).

Учтите, что в графах могут быть кратные рёбра.

Формат выходного файла

На каждую из T пар начальных вершин выведите строку “**first**”, если при правильной игре выиграет первый, и “**second**”, если второй.

Пример

graphprod.in	graphprod.out
3 2	first
1 2	second
2 3	
2 1	
1 2	
2	
1 1	
3 2	

Задача F. Вас снова замаяукали!

Имя входного файла:	labyrinth.in
Имя выходного файла:	labyrinth.in
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Автор задачи:	Denis Denisov

Два котёнка попали в запутанный лабиринт со множеством комнат и переходов между ними. Котят долго по нему плутали, обошли все комнаты по много раз, нашли выход (да даже и не один, а несколько), в общем, изучили там всё, что смогли. Теперь этот лабиринт котята используют в своих играх.

Чаще всего котята играют в следующую игру: начиная в какой-то комнате лабиринта, котята поочерёдно выбирают, в какую из комнат им перейти. Котята изначально находятся в одной комнате и ходят вместе. Как только котёнок, который должен выбрать следующую комнату, не может этого сделать, он признаётся проигравшим. Обычно в таких играх выигрывающий игрок стремится выиграть как можно быстрее, а проигрывающий стремится как можно дольше оттянуть свое поражение. Но у котят свои представления о победе и поражении. Если котёнок знает, что, начиная из текущей комнаты, он выиграет (вне зависимости от действий другого котёнка), то он стремится играть как можно дольше, чтобы продлить себе удовольствие от выигрыша (естественно, при этом выигрывающий котёнок должен гарантировать себе, что будет постоянно уверен в выигрыше). Котёнок, который знает, что проиграет (при условии, конечно, что другой котёнок будет действовать оптимально), старается проиграть как можно быстрее, чтобы начать новую игру, в которой и взять реванш.

Если котята будут ходить бесконечно долго, но никто из них не сможет выиграть, то котята считают игру завершившейся вничью и замаяукивают Вас.

Вас попросили для каждой комнаты в лабиринте узнать, выиграет или проиграет котёнок, начинающий ходить из данной комнаты. Если котёнок, начинающий из этой комнаты, выигрывает, требуется узнать максимальное количество ходов, которое он сможет играть, если же проигрывает — минимальное количество, которое ему придётся играть.

Формат входного файла

В первой строке ввода находятся два числа n и m — число комнат и переходов между комнатами в лабиринте ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$). Далее следует m строк с описаниями переходов. Описание перехода состоит из двух чисел a и b , означающих, что котёнок, начинающий игру в комнате с номером a , может выбрать комнату b в качестве следующей.

Формат выходного файла

Выведите n строк — для каждой комнаты результат игры для котёнка, который начнет игру из этой комнаты. Если игра закончится вничью, выведите «DRAW». Если начинающий котёнок выиграет, выведите «WIN K », где K — количество ходов, которые сможет играть выигрывающий котёнок. Если котёнок сможет играть сколь угодно долго, сохраняя возможность в любой момент выиграть, выведите «WIN INF». Если котёнок, начинающий из этой комнаты, проиграет, выведите «LOSE K », где K — количество ходов, которые придется играть проигрывающему котёнку. Если же котёнку придется играть сколь угодно долго, при том, что его соперник сможет в любой момент выиграть, выведите «LOSE INF».

Примеры

labyrinth.in	labyrinth.in
4 4 1 2 1 3 2 4 3 4	LOSE 2 WIN 1 WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 4 5 5 6	DRAW DRAW DRAW DRAW WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 2 6 4 5	LOSE INF WIN INF LOSE INF WIN INF LOSE 0 LOSE 0