

Содержание

11.Base [2/2]	3
Задача 11A. Одна кучка [0.5 sec, 256 mb]	3
Задача 11B. Ретроанализ для маленьких [1 sec, 256 mb]	4
11.Advanced [3/5]	5
Задача 11C. Короля — в угол 3 [0.3 sec, 256 mb]	5
Задача 11D. Терминатор [0.2 sec, 256 mb]	7
Задача 11E. Жестокая задача [0.5 sec, 256 mb]	8
Задача 11F. Малыш и Карлсон [0.2 sec, 256 mb]	9
Задача 11G. Вариация Нима [0.2 sec, 256 mb]	10
11.Hard [0/3]	11
Задача 11H. Вас снова замяукали! [0.5 sec, 256 mb]	11
Задача 11I. Не-не-не-нечестная игра [0.5 sec, 256 mb]	13
Задача 11J. Игра в пешки [0.5 sec, 256 mb]	14

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/1233/>

Дедлайн на задачи: 2 недели, до 10-го мая 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <https://compscicenter.ru/courses/algorithms-2/2015-spring/>

Семинары ведёт Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии указан таймлит для C/C++.

Таймлит для Java примерно в 2-3 раза больше.

Таймлит для Python примерно в 4-5 раз больше.

11.Base [2/2]

Задача 11А. Одна кучка [0.5 sec, 256 mb]

Два игрока играют в игру. На столе лежит кучка из N камней. Двое ходят по очереди. За ход можно взять a_1, a_2, \dots, a_k камней. Проигрывает тот, кто не может сделать ход. Определите победителя!

Формат входных данных

В первой строке записано число k . Во второй строке k чисел — a_1, a_2, \dots, a_k . В третьей строке идет число m — количество различных N , для каждого из которых требуется определить победителя. В четвертой строке m чисел — N_1, N_2, \dots, N_m .

Ограничения: $1 \leq k \leq 20, m \leq 10^4, 1 \leq N_i, a_i \leq 10^6$.

Формат выходных данных

Выведите m строк, в каждой ответ на вопрос “кто выиграет” — **First** или **Second**.

Пример

heaps2.in	heaps2.out
3	First
1 2 3	First
8	First
1 2 3 4 5 6 7 8	Second
	First
	First
	First
	Second

Задача 11В. Ретроанализ для маленьких [1 sec, 256 mb]

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля в игру. Изначально фишка стоит в вершине i . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Формат входных данных

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой два целых числа n ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n . Пара $a\ b$ обозначает, что ребро ведет из вершины a в вершину b . В графе могут быть петли, могут быть кратные ребра. Сумма n по всем тестам не превосходит 300 000, сумма m по всем тестам также не превосходит 300 000.

Формат выходных данных

Для каждого теста выведите для каждой вершины FIRST, SECOND или DRAW в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Примеры

retro.in	retro.out
5 5 1 2 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 2 2 3 3 1 1 4	?

11.Advanced [3/5]

Задача 11С. Короля — в угол 3 [0.3 sec, 256 mb]

На каждой клетке шахматной доски размеров 8×8 записано целое неотрицательное число. Двое игроков по очереди переставляют короля, перемещая его только вправо, вверх или по диагонали вправо-вверх. Первоначально король стоит в левом нижнем углу. Игра продолжается до тех пор, пока король не окажется в правой верхней клетке доски. Игрок, переставивший короля в некоторую клетку, получает от другого игрока денежную сумму, равную числу, записанному в этой клетке. Определите стоимость игры — сумму, которая окажется в конце игры у первого игрока, если первый игрок старается её максимизировать, а второй — минимизировать.

Формат входных данных

На вход программе подается восемь строк, каждая строка содержит восемь целых неотрицательных чисел, не превосходящих 1000. В левом нижнем углу всегда записано число 0.

Формат выходных данных

Выведите единственное число — стоимость игры.

Пример

king3.in	king3.out
0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0	4

king3.in	king3.out
0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	-3
9 9 9 9 9 9 1 9 9 9 9 9 9 1 9 2 9 9 9 9 9 9 1 9 0 9 9 9 9 9 9 9	9

Задача 11D. Терминатор [0.2 sec, 256 mb]

Два игрока играют в настольную игру. Игровое поле представляет собой квадратный лабиринт, 8×8 клеток. В некоторых клетках располагаются стенки. Один игрок управляет фишкой-терминатором, а второй — фишкой-беглецом. Игроки ходят по очереди, ходы пропускать нельзя (гарантируется, что ход всегда возможен). За один ход игрок может переместить свою фишку в любую из свободных клеток, расположенных рядом с исходной по горизонтали, вертикали или по диагонали (то есть ходом короля). Терминатор, кроме того, может стрелять в беглеца ракетами. Выстрел идет по прямой в любом направлении по горизонтали, вертикали или диагонали. Если беглец оказывается на линии выстрела терминатора и не прикрыт стенками, то терминатор незамедлительно делает выстрел (вне зависимости от того, чей ход), и беглец проигрывает. Начальное положение фишек задано. Первый ход делает беглец. Он выигрывает, если сделает ход с восьмой строки за пределы игрового поля, так как остальные границы поля окружены стенками.

Вопрос задачи: может ли беглец выиграть при оптимальной игре обеих сторон?

Формат входных данных

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — цифрой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

Формат выходных данных

В выходной файл выведите число 1, если беглец выигрывает, и -1 — в противном случае.

Примеры

terminator.in	terminator.out
01000000 10100000 31100000 00020000 00000000 00000000 00000000 00000000	-1

Задача 11Е. Жестокая задача [0.5 sec, 256 mb]

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 входных данных

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

Формат выходных данных

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

cruel.in	cruel.out
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Задача 11F. Малыш и Карлсон [0.2 sec, 256 mb]

На свой День рождения Малыш позвал своего лучшего друга Карлсона. Мама испекла его любимый пирог прямоугольной формы $a \times b \times c$ сантиметров. Карлсон знает, что у Малыша еще есть килограмм колбасы. Чтобы заполучить ее, он предложил поиграть следующим образом: они по очереди разрезают пирог на две ненулевые по объему прямоугольные части с целыми измерениями и съедают меньшую часть (в случае, когда части равные, можно съесть любую). Проигрывает тот, кто не может сделать хода (то есть когда размеры будут $1 \times 1 \times 1$). Естественно, победителю достается колбаса.

Малыш настаивает на том, чтобы он ходил вторым.

Помогите Карлсону выяснить, сможет ли он выиграть, и если сможет — какой должен быть его первый ход для этого.

Считается, что Малыш всегда ходит оптимально.

Формат входных данных

Во входном файле содержится 3 целых числа a, b, c ($1 \leq a, b, c \leq 5\,000$) — размеры пирога.

Формат выходных данных

В случае, если Карлсон не сможет выиграть в Малыша, выведите NO. В противном случае в первой строке выведите YES, во второй — размеры пирога после первого хода Карлсона в том же порядке, что и во входном файле.

Примеры

karlsson.in	karlsson.out
1 1 1	NO
2 1 1	YES 1 1 1

Задача 11G. Вариация Нима [0.2 сек, 256 mb]

На столе лежат n кучек камней: a_1 камней в первой кучке, a_2 камней во второй, \dots , a_n в n -ой. Двое играют в игру, делая ходы по очереди. За один ход игрок может либо взять произвольное ненулевое количество камней (возможно, все) из одной любой кучки, либо произвольным образом разделить любую существующую кучку, в которой не меньше двух камней, на две непустые кучки. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

Формат входных данных

В первой строке задано целое число t — количество тестов ($1 \leq t \leq 100$). Следующие t строк содержат сами тесты. Каждая из них начинается с целого числа n — количества кучек ($1 \leq n \leq 100$). Далее следует n целых чисел a_1, a_2, \dots, a_n через пробел — количество камней в кучках ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите t строк; в i -ой строке выведите “FIRST”, если в i -ом тесте при правильной игре выигрывает первый игрок, и “SECOND”, если второй.

Пример

varnim.in	varnim.out
3	FIRST
1 1	SECOND
2 1 1	FIRST
3 1 2 3	

11.Hard [0/3]

Задача 11Н. Вас снова замаякали! [0.5 sec, 256 mb]

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 котят используют в своих играх.

Чаще всего котят играют в следующую игру: начиная в какой-то комнате лабиринта, котят поочерёдно выбирают, в какую из комнат им перейти. Котят 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 котят будут ходить бесконечно долго, но никто из них не сможет выиграть, то котят считают игру завершившейся вничью и замаякивают 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 входных данных

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 выходных данных

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.out
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

Задача 11I. Не-не-не-нечестная игра [0.5 sec, 256 mb]

Дэвид Блейн и Вася сидят на уроке математики и скучают. Через некоторое время Дэвид Блейн предлагает поиграть в следующую игру:

- он записывает на листочке некоторое большое целое число и ходит первым;
- игроки ходят по очереди;
- на каждом ходу игрок обязан разделить текущее число на его простой делитель p . На этом же ходу игрок может умножить (а может и нет) результат деления на простое число q ($1 < q < p$);
- проигрывает тот, кто не может сделать ход.

Мягко говоря, Вася не очень доверяет Дэвиду Блейну и боится, что тот выписывает только проигрышные для Васи начальные числа. Помогите ему определить, так это или нет.

Формат входных данных

Первая строка входного файла содержит единственное число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

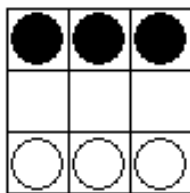
В выходной файл выведите *Vasya*, если у Васи есть выигрышная стратегия, независимо от ходов Дэвида Блейна. Иначе выведите *David*.

Пример

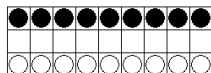
unfair.in	unfair.out
16	Vasya

Задача 11J. Игра в пешки [0.5 sec, 256 mb]

В свободное время Дед Мороз Петрович и Дед Мороз Егорыч играют в следующую игру. На доске 3×3 пешки расставляются следующим образом:



Пешки ходят и бьют по обычным шахматным правилам, к которым добавляется ещё одно: бить обязательно. Проигрывает тот, кто не может сделать ход. Первыми ходят белые. В течение 100 лет Петрович играл белыми и всегда выигрывал. Однажды Егорычу это надоело, и он принёс доску 3×5 . Но и теперь он, играя чёрными, до сих пор проигрывает. «Как так?» — подумал Егорыч и решил купить доску $3 \times N$:



Вот тут Петровичу надо подумать, какими играть, чтобы выиграть. Вам нужно помочь ему это сделать.

Формат входных данных

Целое число N ($1 \leq N \leq 10^9$).

Формат выходных данных

«White», если Петровичу надо играть белыми, и «Black», если надо играть чёрными. Егорыч и Петрович каждый раз ходят по наилучшей для себя стратегии.

Пример

pawns.in	pawns.out
3	White
4	Black
5	White