Общая информация по задачам первого тура

Задача	**Тип задачи**	**Ограничения**
1. Перфокарты	стандартная	1 с, 1024 МБ
2. Родные просторы	стандартная	1 с, 1024 МБ
3. Игра с тайным смыслом	двойной запуск, интерактивная	3 с, 1024 МБ
4. Доклад инвесторам	стандартная	3 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Баллы за подзадачу, если в условии не указано иное, начисляются только если все тесты этой подзадачи пройдены. Решение запускается на тестах для определенной подзадачи, если все тесты всех необходимых подзадач пройдены. Если в задаче за тест могут выставляться частичные баллы, он считается пройденным, если результат работы на нем «OK» или «RS».

Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач указана дополнительно буква У.

Задача 1. Перфокарты

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

На складе фирмы, на базе которой проходит олимпиада по программированию, были обнаружены n перфокарт. Перфокарта представляет собой полоску из m клеточек, каждая из которых либо содержит строчную английскую букву, либо является отверстием.

Жюри олимпиады решило упорядочить все перфокарты так, что если расположить их одну под другой сверху вниз в этом порядке, то получится лозунг олимпиады — заданная строка s длины m.

Иными словами, зафиксируем порядок перфокарт, в котором они будут лежать, и рассмотрим произвольную позицию i ($1 \leqslant i \leqslant m$). Тогда i-й символ строки s должен совпадать с символом на i-й позиции самой верхней перфокарты, содержащей на позиции i какую-либо букву. Если для какого-то i ни одной перфокарты с буквой в позиции i нет, то считается, что требуемую строку s получить невозможно.

Помогите жюри понять, в каком порядке необходимо расположить перфокарты.

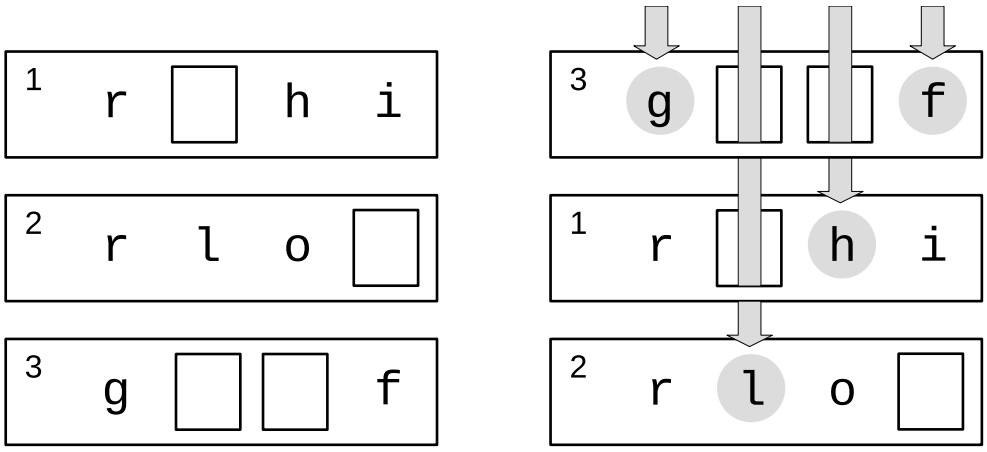


Рис. 1: Порядок карт из второго примера. Выделены те буквы, которые видны сверху

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leqslant n, m \leqslant 100\,000$), обозначающих число перфокарт и количество клеток соответственно.

Вторая строка содержит строку s, состоящую из m строчных английских букв.

В i-й из следующих n строк находится описание i-й перфокарты.

Описание начинается с целого числа k_i ($0 \leqslant k_i \leqslant m$), обозначающего количество позиций с буквами в этой перфокарте. Гарантируется, что сумма всех значений k_i не превышает 200 000.

Далее следует описание букв на этой перфокарте: k_i пар $a_{i,j}$, $c_{i,j}$ ($1 \leqslant a_{i,j} \leqslant m$, $c_{i,j}$ является строчной английской буквой) для всех целых $1 \leqslant j \leqslant k_i$; каждая пара обозначает наличие символа $c_{i,j}$ на позиции $a_{i,j}$. Остальные позиции содержат отверстия. Гарантируется, что номера позиций с буквами для одной перфокарты приведены по возрастанию, то есть для любого $1 \leqslant j < k_i$ верно $a_{i,j} < a_{i,j+1}$.

Формат выходных данных

Если способ упорядочить перфокарты требуемым способом существует, выведите n целых чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$), где p_1 — номер самой верхней перфокарты, p_2 — номер второй сверху перфокарты, и так далее до перфокарты p_n, которая лежит ниже всех. Если возможных ответов несколько, вы можете вывести любой из них.

Если способа упорядочить перфокарты нужным образом не существует, выведите единственное число -1.

Примеры

стандартный ввод	стандартный вывод
1 1 a 1 1 a	1
3 4 glhf 3 1 r 3 h 4 i 3 1 r 2 l 3 o 2 1 g 4 f	3 1 2
2 2 aa 2 1 a 2 b 2 1 b 2 a	-1

Система оценивания

Подзадача	Баллы	Ограничения		Необходимые подзадачи	Информация о проверке
		n	m		
1	15	$n \leqslant 8$	$m \leqslant 100$	У	первая ошибка
2	35	$n \leqslant 100$	$m \leqslant 100$	У, 1	первая ошибка
3	50	$n \leqslant 100\,000$	$m \leqslant 100\,000$	У, 1, 2	первая ошибка

Задача 2. Родные просторы

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Вы играете на смартфоне в игру «Родные просторы», в которой управляющий Остап помогает помещику восстановить отцовский дом. Игра происходит следующим образом.

Дана последовательность из n кристаллов, расположенных в один ряд слева направо. Каждый кристалл относится к одному из k видов, обозначенных первыми k английскими буквами. Таким образом, последовательность кристаллов записывается строкой английских букв.

За один ход игры можно удалить из последовательности один кристалл. Цель игрока — получить в результате применения разрешенных видов удалений лексикографически минимально возможную строку.

Разрешённые виды удаления кристаллов заданы таблицей A размера $k \times k$ из нулей и единиц. Если $A_{ij} = 1$, то разрешается удалить кристалл вида j, если непосредственно слева от него находится кристалл вида i. Данные действия можно выполнять в любом порядке.

Напомним, что строка x лексикографически меньше строки y, если выполнено одно из двух условий:

- существует такая позиция символа m, присутствующая в обеих строках, что до m-го символа строки совпадают, а m-й символ строки x меньше m-го символа y,
- строка x является строгим префиксом y (то есть получается отбрасыванием одного или больше символов с конца строки y).

Формат входных данных

В первой строке даны два целых числа k и n ($1 \leqslant k \leqslant 26$, $1 \leqslant n \leqslant 500\,000$) — количество видов кристаллов и длина исходной последовательности кристаллов.

В следующих k строках задана таблица A, i-я строка содержит ровно k символов 0 или 1. Символ в i-й строке на j-й позиции равен A_{ij}.

В последней строке записаны n строчных английских букв, задающие исходную последовательность кристаллов. Гарантируется, что в строке встречаются только первые k букв английского алфавита, i-я по счёту буква английского алфавита обозначает i-й вид кристаллов.

Формат выходных данных

Выведите лексикографически минимальную строку, которую можно получить из исходной строки разрешёнными действиям.

Примеры

стандартный ввод	стандартный вывод
3 7 010 001 100 abacaba	aac
3 5 010 001 100 bcacb	bacb

Замечание

В примерах из условия разрешены следующие виды удалений (удаляемый символ зачёркнут, символ непосредственно перед ним подчёркнут): a̲b̸, b̲c̸, c̲a̸.

Возможная последовательность удалений в первом примере:

- abacaba
- abaca̲b̸a
- abacaa
- abac̲a̸a
- abaca
- abac̲a̸
- abac
- a̲b̸ac
- aac

Возможная последовательность удалений во втором примере:

- bcacb
- b̲c̸acb
- bacb

Система оценивания

Подзадача	Баллы	Ограничения		Необходимые подзадачи	Информация о проверке
		n	k		
1	10	$n \leqslant 20$	$k \leqslant 26$	У	первая ошибка
2	12	$n \leqslant 50$	$k \leqslant 5$	У	первая ошибка
3	16	$n \leqslant 300$	$k \leqslant 5$	У, 2	первая ошибка
4	17	$n \leqslant 500$	$k \leqslant 26$	У, 1–3	первая ошибка
5	10	$n \leqslant 2\,000$	$k \leqslant 26$	У, 1–4	первая ошибка
6	9	$n \leqslant 10\,000$	$k \leqslant 26$	У, 1–5	первая ошибка
7	8	$n \leqslant 100\,000$	$k \leqslant 26$	У, 1–6	первая ошибка
8	11	$n \leqslant 500\,000$	$k \leqslant 2$		первая ошибка
9	7	$n \leqslant 500\,000$	$k \leqslant 26$	У, 1–8	первая ошибка

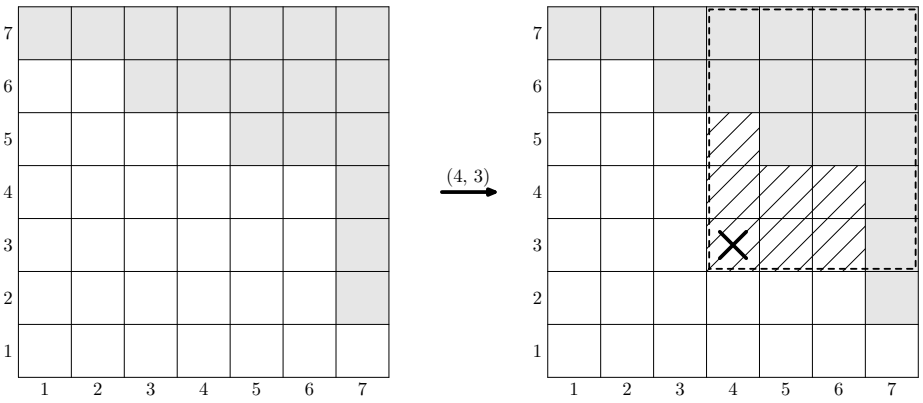
Задача 3. Игра с тайным смыслом

Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Это интерактивная задача с двойным запуском.

Один шпион добыл важное сообщение, и ему необходимо передать его в центр. Так как сообщение важное, было решено, что необходимо скрыть даже сам факт передачи какой-либо информации, и поэтому передача сообщения будет замаскирована под стрим набирающей популярность игры «Щёлк».

Правила игры «Щёлк» очень простые. На квадратном поле размера $n \times n$ изначально ни одна клетка не закрашена. Два игрока ходят по очереди. За один ход игрок может выбрать любую ещё незакрашенную клетку и закрасить её и все незакрашенные клетки прямоугольника, левым нижним углом которого является выбранная клетка, а правым верхним — правый верхний угол поля (см. рисунок). Игрок, закрасивший левый нижний угол поля, проигрывает.



Шпион будет играть на сайте, предоставляющем возможность сыграть с ботом. Бот играет следующим образом: каждый раз он случайно равновероятно выбирает один из ходов, которые закрашивают не более k новых клеток, кроме хода в левый нижний угол. Если же ход в левый нижний угол — единственный оставшийся, то бот делает его и проигрывает игру.

Шпион может сыграть столько игр, сколько необходимо, однако для целей конспирации лучше, чтобы их было как можно меньше. Кроме того, чтобы все выглядело как можно менее подозрительно, лучше как можно больше игр выиграть. С помощью записи ходов, сделанных во всех сыгранных играх, шпион и планирует передать важное сообщение.

Вам необходимо написать программу, которая будет запущена два раза. При первом запуске программа будет исполнять роль шпиона: зная секретное сообщение, она сыграет с ботом несколько игр в «Щёлк», а при втором запуске, получив только списки ходов этих игр, восстановит секретное сообщение.

Протокол взаимодействия

Ваше решение будет запущено два раза.

В первой строке входных данных записано одно число 1 или 2 — номер запуска.

При обоих запусках решения во второй строке записаны три числа: n — размер стороны квадратного поля ($n = 32$ во всех тестах жюри), k — максимальное количество клеток, которые может закрасить бот ($k = 8$ во всех тестах жюри), m — длина сообщения в битах ($m = 100\,000$ во всех тестах, кроме примера).

При первом запуске в третьей строке находится битовое представление передаваемого сообщения — строка длины m, состоящая из символов «`0`» и «`1`». После его считывания решение должно начать взаимодействие с программой жюри, реализующей стратегию бота. Взаимодействие состоит в обмене ходами. Ход задается парой чисел — номерами столбца и строки выбранной клетки соответственно. Столбцы нумеруются слева направо от 1 до n, а строки — снизу вверх от 1 до n. Выбранная клетка не должна быть закрашенной, иначе решение будет прервано и получит вердикт «`WA`». После того, как один из игроков сделал проигрышный ход «`1 1`», игра заканчивается, и тут же

начинается новая. В первой игре первый ход делает ваша программа, а далее первый ход чередуется, то есть во второй игре первой будет ходить программа жюри, в третьей снова ваша программа и так далее.

Когда решение считает, что записи имеющихся игр достаточно для передачи сообщения, вместо своего первого хода в игре необходимо вывести единственное число 0, и завершить работу решения. Даже если в этой игре первый ход делает бот, можно вывести 0 вместо своего первого хода и завершить работу решения. В таком случае первый ход бота не будет записан в лог, доступный решению на втором запуске. В середине игры прервать ее нельзя.

Если решение во время первого запуска сыграет с соблюдением протокола некоторое число игр, уложившись в ограничения по времени и памяти, оно будет запущено второй раз. Иначе второго запуска не будет, и вердикт при первом запуске («WA», «RE», «TL», «ML» или «IL») будет объявлен результатом проверки на этом тесте.

При втором запуске в третьей строке будет находиться количество игр, которые решение сыграло на первом запуске. После этого на вход будет подан список ходов сделанных во всех играх, как решением, так и ботом, в порядке, в котором эти ходы происходили. При втором запуске решение должно вывести одну строку длины m из «0» и «1» — секретное сообщение, которое было доступно первому запуску.

При втором запуске вам необходимо сначала считать все данные из стандартного потока ввода, и только затем вывести ответ.

Если второй запуск состоялся, и вердикт второго запуска «RE», «TL», «ML» или «IL», — этот результат будет объявлен результатом проверки на этом тесте. Иначе выведенный программой ответ будет проверен. Если он совпадает с переданным программе при первом запуске секретным сообщением, результат проверки на тесте будет «OK», иначе — «WA». В обоих запусках после каждого вывода вашей программы выводите перевод строки и делайте сброс потока вывода.

Если вы используете «cout << ... << endl» в C++, «System.out.println» в Java, «print» в Python, «Console.WriteLine» в C#, «writeln» в Паскале, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса потока вывода можно использовать «fflush(stdout)» в C и C++, «flush(output)» в Паскале, «System.out.flush()» в Java, «sys.stdout.flush()» в Python, «Console.Out.Flush()» в C#.

Для удобства локального тестирования, вы можете использовать вариант программы жюри с визуализацией игры, доступной в рабочем каталоге. Программу можно запустить командой «java -jar visualizer.jar» из консоли, или двойным кликом по файлу в файловом менеджере. Вы можете также скачать визуализатор в тестирующей системе на вкладке «Файлы».

Ваше решение может взаимодействовать с программой жюри, используя для ввода-вывода файлы, которые будут указаны после нажатия на кнопку «Начать игру». У решения может не получиться открыть файлы до того, как запущена программа жюри, поэтому следует нажимать кнопку «Начать игру» и только после этого запускать решение. После завершения взаимодействия будет предложено сохранить лог игры в формате входных данных для второго запуска и секретное сообщение, чтобы вы могли проверить корректность его восстановления вашей программой. Кроме проверки взаимодействия выданная вам программа позволяет посмотреть как проходила игра между вашим решением и программой жюри. Для этого нужно перед запуском поставить настройку «Визуализировать игру». Также можно сыграть с программой жюри самостоятельно, выбрав соответствующую опцию в интерфейсе.

Система оценивания

Ваше решение запускается на 10 тестах, за каждый из которых можно получить до 10 баллов.

Секретное сообщение в каждом тесте зафиксировано и имеет длину $m = 100\,000$. В каждом тесте сообщение было получено генератором случайных чисел, каждый символ независимо с равной вероятностью был выбран равным «0» или «1».

Ходы, которые делает решение жюри в каждом тесте, случайны, в соответствии с описанием в условии, но при повторных запусках, если программа участника будет делать те же самые ходы, то программа жюри также будет делать те же самые ходы в этом тесте.

Если решение не смогло правильно восстановить сообщение на втором запуске или нарушило протокол взаимодействия на первом запуске, за тест будет выставлено 0 баллов.

Иначе балл за тест зависит от количества сыгранных игр при первом запуске, а также от количества игр, которые были выиграны программой-решением.

Пусть $W = 1$, если проиграно не более одной игры, и $W = 0$ в противном случае. Пусть G — количество сыгранных игр, $B = \frac{m}{G}$ — среднее количество бит, переданных за одну игру. Числа b_i определяют значение B, необходимое для получения i баллов, и представлены в таблице ниже.

b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
0	1	100	300	450	600	700	750	800	825

В случае если $B \geqslant b_9$, решение получает $W + 9$ баллов за тест. Иначе, пусть s таково, что $b_s \leqslant B < b_{s+1}$. В таком случае, решение получит $W + s + \frac{B - b_s}{b_{s+1} - b_s}$ баллов за тест. Баллы за отдельные тесты будут просуммированы, и уже итоговая сумма будет округлена до ближайшего целого числа.

Вашим результатом по этой задаче будет результат наилучшей попытки в целом.

Пример

Приведенный ниже пример предназначен только для того, чтобы ознакомиться с форматом ввода и вывода. В частности, не предполагается, что в указанном примере шпион действительно передал какую-либо информацию с помощью приведенной там последовательности ходов.

Ваше решение не будет запущено на тесте из примера при отправке в тестирующую систему.

Первый запуск.

стандартный ввод	стандартный вывод
1 32 8 16 1111000011110000 30 1 28 1 1 1 31 31 	 3 2 1 2 2 1 0

В примере ввод и вывод отформатированы пустыми строками, чтобы было видно, какие запросы соответствуют каким ответам программы жюри. В реальном взаимодействии необходимо переводить строку после каждого запроса, но выводить пустые строки не надо.

Второй запуск.

стандартный ввод	стандартный вывод
2 32 8 16 1 3 2 30 1 1 2 28 1 2 1 1 1	1111000011110000

Задача 4. Доклад инвесторам

Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

В компании работают n бурундуков ($n \geqslant 2$), пронумерованных целыми числами от 1 до n. Бурундук с номером 1 является основателем компании, а каждый из остальных бурундуков имеет ровно одного непосредственного начальника. Иными словами, иерархия бурундуков представляет собой корневое дерево, где родитель вершины является её начальником, а дети вершины являются её подчинёнными. Бурундуки, имеющие подчинённых, называются *менеджерами*, а остальные — *консультантами*. Структура компании такова, что каждый менеджер имеет не более 8 подчинённых. Обратите внимание, что основатель компании также является менеджером.

Основатель компании решил составить доклад для инвесторов о последних проделанных улучшениях разрабатываемого продукта. Каждое улучшение является результатом работы кого-то из консультантов компании. Все улучшения пронумерованы в хронологическим порядке их совершения.

Для каждого из консультантов известен список улучшений, сделанных этим консультантом. Каждый консультант обязан выбрать одно из этих улучшений и сделать о нём доклад своему менеджеру. Таким образом, доклад любого консультанта будет состоять ровно из одного улучшения.

Каждый менеджер, в том числе основатель компании, должен сделать доклад, представляющий собой объединение докладов всех его подчиненных. Для этого он берёт доклады, полученные от подчинённых, и записывает их подряд без изменений в некотором порядке. Например, если первый доклад состоит из улучшений $[1, 3]$, а второй — из улучшений $[2, 4, 10]$, то в результате объединения может получиться доклад $[2, 4, 10, 1, 3]$, либо доклад $[1, 3, 2, 4, 10]$, никакие другие доклады получиться не могут.

Основатель компании стремится рассказать об улучшениях максимально логично, поэтому он хочет, чтобы в его докладе улучшения шли в хронологическом порядке, то есть по возрастанию номеров. Помогите консультантам выбрать, о каком улучшении они должны рассказать, а менеджерам выбрать, в каком порядке им располагать доклады подчиненных при объединении, чтобы в итоговом докладе основателя компании вошедшие в него улучшения шли в хронологическом порядке.

Формат входных данных

В первой строке содержится целое число n ($2 \leqslant n \leqslant 100\,000$) — количество бурундуков в компании. Далее следуют n описаний бурундуков, по одному в строке:

- если бурундук является менеджером, то описание начинается с числа 1, затем следует число k_i — количество непосредственных подчинённых у этого бурундука ($1 \leqslant k_i \leqslant 8$), а затем следуют k_i различных чисел от 2 до n — номера бурундуков, которые являются его подчинёнными;
- если бурундук является консультантом, то описание начинается с числа 2, затем следует число m_i — количество улучшений, о которых этот консультант может рассказать, а затем следуют m_i различных целых чисел от 1 до 100 000 — номера улучшений.

Бурундук с номером 1 (основатель компании) всегда является менеджером, а каждый из остальных бурундуков упомянут в качестве подчинённого ровно один раз и прямо или косвенно подчинен основателю компании.

Сумма чисел m_i по всем консультантам не превосходит 100 000. Гарантируется, что никакое улучшение не было сделано двумя различными консультантами, то есть все упомянутые всеми консультантами улучшения различны.

Формат выходных данных

Если составить доклад требуемым образом невозможно, выведите «`No`».

Если доклад составить возможно, выведите «`Yes`». В этом случае затем вы можете вывести *сертификат*, который описывает для каждого из бурундуков в порядке их номеров следующее:

- если бурундук является менеджером, то нужно вывести список его подчиненных в том порядке, в котором требуется расположить их доклады;
- если бурундук является консультантом, то нужно вывести номер улучшения, о котором ему нужно рассказать в своем докладе.

Особенность этой задачи заключается в следующем: вам разрешается как выводить сертификат, так и не выводить его. Если решение не выводит сертификат, но правильно определяет возможность составления доклада, оно получит часть баллов за тест.

Обратите внимание, что вывод некорректного сертификата приводит к вердикту «`Неправильный ответ`» и нулю баллов за этот тест, несмотря на правильность определения возможности составления доклада.

Примеры

стандартный ввод	стандартный вывод
6 1 3 5 4 6 2 3 10 61 60 2 2 80 20 2 2 40 70 1 2 3 2 2 4 30 90 91 92	Yes 5 6 4 10 20 40 2 3 30
3 1 2 2 3 2 1 1 2 1 2	Yes
5 1 2 2 3 2 1 2 1 2 4 5 2 1 1 2 1 3	No

Замечание

Во втором примере не сделан вывод сертификата, что соответствует частичному решению.

В третьем примере каждый из консультантов сделал ровно одно улучшение, поэтому выбор улучшений однозначен.

У третьего менеджера есть два возможных варианта доклада: $[1, 3]$ или $[3, 1]$.

У первого менеджера есть четыре возможных варианта доклада с учетом всех возможностей доклада третьего менеджера: $[1, 3] + [2] = [1, 3, 2]$, $[2] + [1, 3] = [2, 1, 3]$, $[3, 1] + [2] = [3, 1, 2]$ и $[2] + [3, 1] = [2, 3, 1]$, но ни в одном из этих докладов улучшения не идут в хронологическом порядке.

Система оценивания

Если во всех тестах подзадачи правильно определена возможность составления доклада, а также во всех тестах, где доклад возможен, предъявлен верный сертификат, за подзадачу выставляется полный балл.

В противном случае, если во всех тестах подзадачи правильно определена возможность составления доклада, и на любой тест сертификат либо верный, либо отсутствует, за подзадачу выставляется частичный балл. Обратите внимание, что подзадачи, зависящие от неё, в таком случае будут запущены для тестирования и даже могут быть оценены в полный балл.

Обозначим за K максимальное количество непосредственных подчинённых у бурундуков-менеджеров, т.е. $K = \max k_i$. Также обозначим суммарное количество улучшений у бурундуков-консультатов за $\sum m_i$.

Подз.	Баллы		Ограничения		Необх. подзадачи	Информация о проверке
	Частичные	Полные	n, $\sum m_i$	K		
1	9	18	$n, \sum m_i \leqslant 10$	$K \leqslant 2$	–	первая ошибка
2	3	6	$n, \sum m_i \leqslant 20$	$K \leqslant 8$	У, 1	первая ошибка
3	2	4	$n, \sum m_i \leqslant 100$	$K \leqslant 2$	1	первая ошибка
4	2	4	$n, \sum m_i \leqslant 100$	$K \leqslant 5$	У, 1, 3	первая ошибка
5	2	4	$n, \sum m_i \leqslant 100$	$K \leqslant 8$	У, 1–4	первая ошибка
6	2	4	$n, \sum m_i \leqslant 500$	$K \leqslant 2$	1, 3	только баллы
7	2	4	$n, \sum m_i \leqslant 500$	$K \leqslant 5$	У, 1, 3, 4, 6	только баллы
8	2	4	$n, \sum m_i \leqslant 500$	$K \leqslant 8$	У, 1–7	только баллы
9	4	8	$n, \sum m_i \leqslant 2000$	$K \leqslant 2$	1, 3, 6	только баллы
10	3	6	$n, \sum m_i \leqslant 2000$	$K \leqslant 5$	У, 1, 3, 4, 6, 7, 9	только баллы
11	3	6	$n, \sum m_i \leqslant 2000$	$K \leqslant 8$	У, 1–10	только баллы
12	6	12	$n, \sum m_i \leqslant 100\,000$	$K \leqslant 2$	1, 3, 6, 9	только баллы
13	3	6	$n, \sum m_i \leqslant 100\,000$	$K \leqslant 5$	У, 1, 3, 4, 6, 7, 9, 10, 12	только баллы
14	7	14	$n, \sum m_i \leqslant 100\,000$	$K \leqslant 8$	У, 1–13	только баллы