



Общая информация по задачам олимпиады

Ограничение по памяти

- В задаче A: 256 мегабайт
- В задаче B: 256 мегабайт
- В задаче C: 256 мегабайт
- В задаче D: 256 мегабайт
- В задаче E: 1024 мегабайт

Ограничение на размер исходного кода программы

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Ограничение на посылку решений

По каждой задаче на проверку принимается не более 50 решений.

По каждой задаче участник не может отправить решение более одного раза в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

Система оценки

Каждая задача олимпиады поделена на несколько подзадач. Чтобы набрать баллы по подзадаче, программа должна пройти все тесты этой подзадачи.

За каждую задачу выставляется суммарный балл по всем ее подзадачам. В каждой подзадаче оценивается лучшее решение, то есть за подзадачу выставляется максимальный набранный по ней балл среди всех решений.

Получение информации о результатах проверки

Чтобы получить информацию о проверке вашего решения, используйте ссылку «Информация о проверке» во вкладке «Решения» в PCMS2 Web Client. По каждой задаче вам будет доступна информация по количеству набранных баллов в каждой подзадаче или результат проверки на первом непройденном тесте.

Таблица результатов

Во время проведения тура олимпиады таблица результатов будет недоступна для просмотра участниками. Официальные результаты будут опубликованы после завершения тура в установленном организаторами порядке.



Задача А. Свидание

Ограничение по времени: 1 секунда

Ринат и Ляйсан — два белых барса. Ринат давно хотел пригласить Ляйсан на свидание. Но оказалось, что найти подходящее время для встречи не так просто: у обоих очень необычные расписания.

Начиная с момента времени 0, Ринат первые a секунд свободен, следующие a секунд занят, затем снова a секунд свободен и так далее.

У Ляйсан расписание устроено наоборот: первые b секунд она занята, следующие b секунд свободна, затем снова b секунд занята и так далее.

Чтобы пригласить Ляйсан на свидание, Ринату нужно найти момент, начиная с которого они оба будут свободны в течение хотя бы одной полной секунды.

Ринат готов ждать не дольше момента времени T .

Найдите минимальное целое число t ($0 \leq t \leq T$), такое, что Ринат и Ляйсан оба свободны в течение всего промежутка $[t, t + 1)$, то есть с момента t включительно до момента $t + 1$ не включительно.

Если такого момента не существует, выведите -1 .

Формат входных данных

В первой строке даны три целых числа a , b и T ($1 \leq a, b \leq 10^{18}$, $0 \leq T \leq 10^{18}$).

Формат выходных данных

Выведите минимальное целое число t , такое, что в течение всего промежутка времени $[t, t + 1)$ Ринат и Ляйсан оба свободны.

Если такого t не существует, выведите -1 .

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы, всех тестов необходимых для неё групп и тестов из условия.

Группа	Баллы	Дополнительные ограничения	Необходимые группы
1	15	$T \leq 10^6$	—
2	15	$a, b \leq 10^3$	—
3	10	$b < a$	—
4	15	$b < 2a$	3
5	15	$b < 1\,000\,001 \cdot a$	2–4
6	10	b делится на a без остатка	—
7	20	Нет дополнительных ограничений	1–6

Примеры

стандартный ввод	стандартный вывод
5 3 10	3
3 5 10	6
3 8 5	-1



Замечание

В первом примере Ринат свободен в течение первых 5 секунд, то есть на промежутке $[0, 5)$, затем занят на $[5, 10)$. Ляйсан, наоборот, занята на $[0, 3)$ и свободна на $[3, 6)$. Поэтому впервые они одновременно свободны на промежутке $[3, 5)$, и минимальный подходящий момент равен 3.

Во втором примере Ляйсан свободна на $[5, 10)$. В момент 5 Ринат ещё занят, так как его занятый промежуток — $[3, 6)$. Начиная с момента 6 он снова свободен, поэтому ответ равен 6.

В третьем примере до момента $T = 5$ подходящего момента нет: Ляйсан становится свободна только в момент 8. Поэтому ответ равен -1 .



Задача В. Вторая попытка

Ограничение по времени: 2 секунды

Несмотря на все старания, Ляйсан всё-таки отказала Ринату. Но Ринат не привык сдаваться!

Чтобы произвести на Ляйсан впечатление, Ринат решил подарить ей букет цветов. Он купил n цветов и перед тем, как собрать букет, разложил их в ряд. Каждый цветок относится к некоторому виду. Для каждого цветка Ринат записал число a_i — номер вида i -го цветка в ряду.

Однако Ринат узнал, что Ляйсан любит разнообразие и хочет, чтобы в букете не было двух цветов одного вида. Иными словами, номера видов всех оставшихся цветов должны быть попарно различными.

Менять порядок цветов Ринат не хочет. Он может лишь убрать один непрерывный фрагмент ряда: выбрать два индекса l и r ($1 \leq l \leq r \leq n$) и убрать цветы, стоящие на позициях $l, l+1, \dots, r$.

Если номера видов всех цветов уже попарно различны, Ринат может ничего не убирать. В таком случае считается, что он убрал 0 цветов.

Помогите Ринату определить минимальное количество подряд стоящих цветов, которые необходимо убрать, чтобы номера видов всех оставшихся цветов были попарно различными.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($1 \leq n \leq 10^6$) — количество цветов в ряду.

Вторая строка каждого набора содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — номера видов цветов.

Гарантируется, что сумма n по всем наборам входных данных не превышает 10^6 .

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимальное количество подряд идущих цветов, которые необходимо убрать, чтобы номера видов всех оставшихся цветов были попарно различными.



Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы, всех тестов необходимых для неё групп и тестов из условия.

Группа	Баллы	Дополнительные ограничения	Необходимые группы
1	10	$n \leq 100$	–
2	5	$n \leq 10^3$	1
3	10	$n \leq 5 \cdot 10^3$	1, 2
4	10	$n \leq 10^4$	1–3
5	15	Каждое число встречается не более двух раз, $n \leq 10^5$	–
6	15	$n \leq 5 \cdot 10^4$	1–4
7	10	$n \leq 10^5$	1–6
8	10	$n \leq 5 \cdot 10^5$	1–7
9	15	Нет дополнительных ограничений	1–8

Пример

стандартный ввод	стандартный вывод
5	2
5	0
1 2 1 3 2	6
6	2
0 1 2 3 4 5	1
7	
4 4 4 4 4 4 4	
8	
1 2 3 2 4 5 3 6	
6	
5 1 2 3 1 4	

Замечание

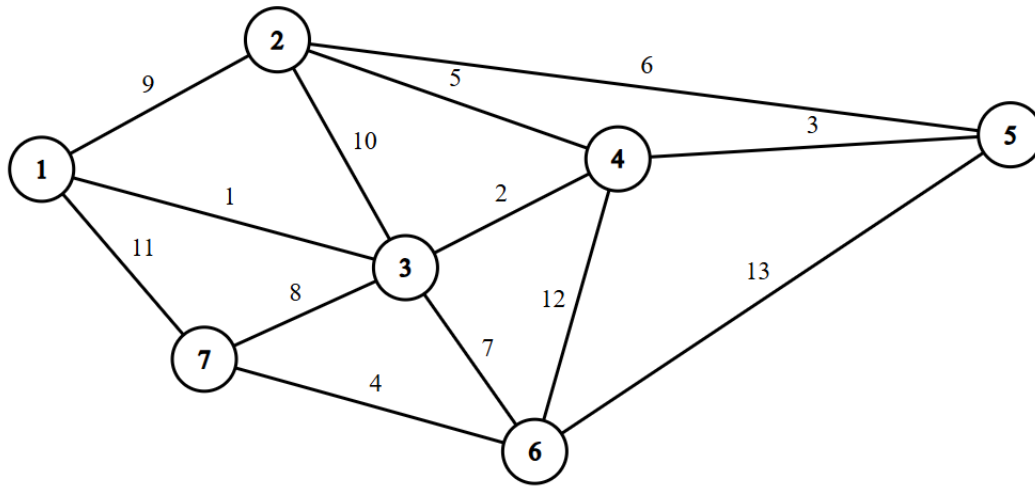
- В первом наборе можно удалить фрагмент [2, 3].
- Во втором наборе ничего удалять не требуется.
- В третьем наборе можно удалить фрагмент [1, 6].
- В четвёртом наборе можно удалить фрагмент [3, 4].
- В пятом наборе можно удалить фрагмент [2, 2].



Задача С. Лесные тропы

Ограничение по времени: 1 секунда

Вы попали в гости к местному лесничему и увидели схематичную карту местности.



На карте можно наблюдать пронумерованные от 1 до 7 перекрёстки троп (или перепутья) и сами лесные тропы, пронумерованные от 1 до 13.

Здесьние лесники придумали свой алгоритм обхода леса и записали его в виде функции ЛЕС:

```
ЛЕС(П, Р)
рас[П] = Р
для всех троп вида (П, С) или (С, П)
если рас[С] не задано
    ЛЕС(С, рас[П] + 1)
иначе
    рас[П] = мин(рас[П], рас[С] + 1)
```

В данной функции:

- `рас` — массив расстояний, значения которого изначально не заданы;
- `мин` — функция минимума;
- `для всех троп` — перебор всех троп;
- рассматриваются тропы **всегда в порядке возрастания** номера тропы.

Подробно о том, как данная функция отработает на карте местности, можно найти в **замечании**.

Лесники постоянно используют этот алгоритм при обходе леса, опуская расчёт расстояний. Но салаги (лесники, впервые обходящие лес) используют его в том числе для определения минимального расстояния (точнее минимального количества троп на пути — будем считать длины всех троп равными 1) от главного входа (перекрёсток 1) до каждого из перекрёстков — фактически салага является исполнителем вызова функции `ЛЕС(1, 0)`.



Взглянув на эту функцию, вы и местный лесничий поняли, что она не всегда находит минимальные расстояния верно. Чтобы убедить лесников перестать ей пользоваться для определения расстояний, вы задумали предъявить t примеров карт местности, в i -м из которых n_i перепутий, но только у k_i из них расстояние было определено верно. В некоторых примерах количество троп тоже уже выбрано и равно m_i . Подробно о других ограничениях на примеры — в **формате выходных данных**.

Формат входных данных

В первой строке находится одно целое положительное число t ($1 \leq t \leq 500$) — количество примеров.

В i -й из следующих t строк содержится три целых числа n_i , m_i и k_i — ограничения на i -й пример ($1 \leq n_i \leq 300\,000$; $0 \leq k_i \leq n_i$; $-1 \leq m_i \leq \min\left(\frac{n_i \cdot (n_i - 1)}{2}, 300\,000\right)$). Если $m_i = -1$, то количество троп в примере ещё не выбрано.

Гарантируется, что сумма всех w_i не превосходит 300 000, где $w_i = m_i$, если $m_i \neq -1$, а иначе $w_i = \min\left(\frac{n_i \cdot (n_i - 1)}{2}, 300\,000\right)$.

Формат выходных данных

Выведите очередной ответ на просьбу привести пример с новой строки в следующем формате:

- Выведите «Ok m'_i » (без кавычек) и с новой строки описание m'_i троп, если пример существует, причём $m'_i = m_i$, если $m_i \neq -1$, а иначе $0 \leq m'_i \leq \min\left(\frac{n_i \cdot (n_i - 1)}{2}, 300\,000\right)$;
- Выведите «Impossible» (без кавычек), если примера не существует.

Если пример существует, то в j -й из следующих m'_i строк выведите описание j -й тропы: два числа u_j и v_j — номера перекрёстков, соединённых j -й тропой.

По тропам должно быть возможно дойти от любого перекрёстка до любого, причём каждая пара перекрёстков соединена не более чем одной тропой. Разумеется, каждая тропа соединяет ровно два различных перекрёстка и ходить по ней можно в обоих направлениях.

В остальных аспектах предоставленные вами примеры не обязаны быть реалистичными. Например, может не существовать ни одного способа нарисовать предоставленный вами пример без пересечения троп и это не будет считаться за неподходящий пример — просто на некоторых перекрёстках не так просто свернуть с пути.



Система оценки

Тесты к этой задаче состоят из тринадцати групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых для неё групп.

Группа	Баллы	Дополнительные ограничения	Необходимые группы
0	0	Тесты из условия	–
1	8	$n_i \leq 8$ и $m_i = -1$	–
2	12	$n_i \leq 5000$ и $m_i = -1$	1
3	8	$0 \leq m_i < n_i - 1$	–
4	13	$k_i \leq 3$	–
5	6	$k_i = n_i$	–
6	10	$n_i \leq 8$	0, 1
7	10	$n_i \leq 100$	0, 1, 6
8	10	$n_i \leq 5000$ и $m_i = n_i$	–
9	4	$n_i \leq 5000$ и $m_i \geq n_i + k_i$	0
10	4	$n_i \leq 5000$	0–2, 6–9

В группах 11–13 осуществляется **offline-проверка** — это означает, что результаты тестирования вашего решения на данных группах станут доступны только после окончания тура.

Группа	Баллы	Дополнительные ограничения	Необходимые группы
11	5	$m_i = -1$	1, 2
12	5	$m_i = n_i$	8
13	5	–	0–12

Пример

стандартный ввод	стандартный вывод
1 7 13 6	Ok 13 1 3 4 3 4 5 6 7 2 4 2 5 3 6 7 3 1 2 2 3 1 7 4 6 5 6



Замечание

```
ЛЕС(1, 0)
рас[1] = 0
тропа 1 (1, 3): вызов ЛЕС(3, 1)
ЛЕС(3, 1)
рас[3] = 1
тропа 1 (1, 3): рас[3] не изменилось
тропа 2 (4, 3): вызов ЛЕС(4, 2)
ЛЕС(4, 2)
рас[4] = 2
тропа 2 (4, 3): рас[4] не изменилось
тропа 3 (4, 5): вызов ЛЕС(5, 3)
ЛЕС(5, 3)
рас[5] = 3
тропа 3 (4, 5): рас[5] не изменилось
тропа 6 (2, 5): вызов ЛЕС(2, 4)
ЛЕС(2, 4)
рас[2] = 4
тропа 5 (2, 4): рас[2] = мин(рас[2], рас[4] + 1) = мин(4, 2 + 1) = 3
тропа 6 (2, 5): рас[2] не изменилось
тропа 9 (1, 2): рас[2] = мин(рас[2], рас[1] + 1) = мин(3, 0 + 1) = 1
тропа 10 (2, 3): рас[2] не изменилось
тропа 13 (5, 6): вызов ЛЕС(6, 4)
ЛЕС(6, 4)
рас[6] = 4
тропа 4 (6, 7): вызов ЛЕС(7, 5)
ЛЕС(7, 5)
рас[7] = 5
тропа 4 (6, 7): рас[7] не изменилось
тропа 8 (7, 3): рас[7] = мин(рас[7], рас[3] + 1) = мин(5, 1 + 1) = 2
тропа 11 (1, 7): рас[7] = мин(рас[7], рас[1] + 1) = мин(2, 0 + 1) = 1
тропа 7 (3, 6): рас[6] = мин(рас[6], рас[3] + 1) = мин(4, 1 + 1) = 2
тропа 12 (4, 6): рас[6] не изменилось
тропа 13 (5, 6): рас[6] не изменилось
тропа 5 (2, 4): рас[4] не изменилось
тропа 12 (4, 6): рас[4] не изменилось
тропа 7 (3, 6): рас[3] не изменилось
тропа 8 (7, 3): рас[3] не изменилось
тропа 10 (2, 3): рас[3] не изменилось
тропа 9 (1, 2): рас[1] не изменилось
тропа 11 (1, 7): рас[1] не изменилось
```

Итого:

Перепутье	1	2	3	4	5	6	7
Расстояние	0	1	1	2	3	2	1

Единственное неверное расстояние — расстояние до перепутья 5.



Задача D. F1

Ограничение по времени: 2 секунды

Кот Miston нашёл на чердаке фрагмент старой советской газеты о гонке F1. На полях было написано $2m$ чисел, по два числа в строке.

По другим заметкам Miston понял, что значат эти числа: кто-то профессионально, в хронологическом порядке застенографировал все обгоны. То есть числа f_i и s_i в i -й строке значили, что гонщик номер f_i обогнал гонщика с номером s_i . Причём записаны только те обгоны, что изменяли таблицу результатов (иными словами, обгоны на круг или несколько кругов записаны не были).

Однако нигде нет описания стартовой решётки — это порядок, в котором гонщики выстраиваются перед началом гонки. Нашлась лишь запись об общем количестве гонщиков — их было n .

После покидания пыльного чердака котята завалили уставшего кота вопросами вида «мог ли гонщик x_i в итоге оказаться на месте p_i ?». Придётся отвечать.

Формат входных данных

В первой строке даны три целых числа n , m и q ($0 \leq n, m, q \leq 3 \cdot 10^5$; $n \neq 0$; $q \neq 0$) — количество гонщиков, обгонов и вопросов.

В i -й из следующих m строк даны два целых числа f_i и s_i ($1 \leq f_i, s_i \leq n$; $f_i \neq s_i$) — числа из i -й строки на полях.

Гарантируется, что данные об обгонах корректны, то есть существует хотя бы одна такая стартовая решётка, что все записанные обгоны осуществимы в указанном порядке.

В i -й из следующих q строк даны два целых числа x_i и p_i ($1 \leq x_i, p_i \leq n$) — номер гонщика и место из i -го вопроса.

Формат выходных данных

Выведите q строк, в i -й из которых должно быть написано «Yes» (без кавычек), если ответ на i -й вопрос положительный, а иначе — «No» (без кавычек).

Считайте, что все гонщики доехали до финиша, никого не дисквалифицировали и никто не получил отложенных штрафов по времени.



Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых для неё группы.

Группа	Баллы	Дополнительные ограничения	Необходимые группы
0	0	Тесты из условия	–
1	5	$m = 0$	–
2	17	$n, m \leq 10$	0
3	8	Лидер финишировал последним*	–
4	7	$p_i = 1$	–
5	11	$p_i \leq 3$	4
6	12	$n, m \leq 1000$	0, 2
7	11	$n \leq 20\,000$	0, 2, 6
8	7	$q \leq 10$	–
9	22	–	0–8

* Гонщик на поул-позиции, то есть первый на старте, пришёл к финишу последним.

Пример

стандартный ввод	стандартный вывод
6 4 6 4 2 2 4 2 3 5 1 4 2 1 2 3 5 1 1 5 5 6 1	No Yes Yes No Yes Yes

Замечание

Рассмотрим все подходящие к примеру стартовые решётки и соответствующие им результаты гонки и ответы на вопросы.

Стартовая решётка	Порядок финиша	1	2	3	4	5	6
1, 5, 3, 2, 4, 6	5, 1, 2, 3, 4, 6	No	Yes	No	No	No	No
1, 5, 6, 3, 2, 4	5, 1, 6, 2, 3, 4	No	Yes	Yes	No	No	No
3, 2, 4, 1, 5, 6	2, 3, 4, 5, 1, 6	No	No	No	No	No	No
3, 2, 4, 6, 1, 5	2, 3, 4, 6, 5, 1	No	No	No	No	Yes	No
6, 1, 5, 3, 2, 4	6, 5, 1, 2, 3, 4	No	No	Yes	No	No	Yes
6, 3, 2, 4, 1, 5	6, 2, 3, 4, 5, 1	No	No	No	No	Yes	Yes



Задача Е. Быть шулером

Ограничение по времени: 5 секунд

Рассмотрим шоу-викторину со следующими правилами: есть n участников и k тем для вопросов, пронумерованных от 1 до k . У каждого игрока есть сильная тема a_i и слабая тема b_i . Конечно, $a_i \neq b_i$. Каждый игрок на вопрос по своей сильной теме всегда отвечает верно, на вопрос по слабой теме — всегда неверно, а на вопрос по любой из средних тем отвечает верно на первые $L - 1$ вопросов, а начиная с L -го вопроса — неверно.

Однако не всё так просто. Среди игроков затаился один шулер! Шулер всегда отвечает верно на все вопросы, игнорируя предыдущие правила.

Вам, как ведущему, хочется вычислить шулера за минимальное число вопросов. Для каждого вопроса вы выбираете тему, и этот вопрос задаётся всем участникам. После чего, вы узнаете кто ответил на вопрос верно, а кто неверно. После этого вы можете задать следующий вопрос. Вам известно, что ровно один из игроков является шулером.

Пусть минимальное число вопросов, которое необходимо задать, чтобы гарантированно вычислить шулера для игры с участниками под номерами $l, l + 1, \dots, r$ это $f(l, r)$.

Вычислите $\sum_{l=1}^n \sum_{r=l}^n f(l, r)$. Иными словами, рассмотрите все возможные непрерывные отрезки участников с номерами от l до r , где $1 \leq l \leq r \leq n$. Для каждого такого отрезка определите минимальное количество вопросов, необходимое для гарантированного определения шулера, если в игре участвуют только игроки с номерами от l до r .

В качестве ответа найдите сумму этих количеств по всем таким отрезкам.

Формат входных данных

Первая строка содержит три целых числа n , k и L ($2 \leq n \leq 200\,000$, $2 \leq k \leq n$, $2 \leq L \leq n$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq k$).

Третья строка содержит n целых чисел $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq k$).

Для каждого i выполнено $a_i \neq b_i$.

Формат выходных данных

Выведите одно целое число — сумму значений $f(l, r)$ по всем парам индексов l и r , удовлетворяющим условию $1 \leq l \leq r \leq n$.

Система оценки

Тесты задачи разделены на 16 подгрупп. Баллы за подгруппу начисляются только при прохождении всех её тестов и всех тестов необходимых подгрупп, указанных в таблице. В таблице приведены дополнительные ограничения; остальные ограничения совпадают с общими.



Подгруппа	Баллы	Дополнительные ограничения			Необходимые подгруппы
		n	k	Прочие условия	
0	0	Примеры из условия			–
1	2	–	–	$b_i = 1$ для всех i	–
2	7	$n \leq 7$	–	–	–
3	9	$n \leq 70$	–	–	0, 2
4	5	$n \leq 3000$	$k \leq 8$	–	0, 2
5	6	$n \leq 3000$	$k \leq 30$	–	0, 2, 4
6	7	$n \leq 3000$	–	–	0, 2–5
7	4	–	$k = 2$	–	–
8	5	–	$k = 3$	$L = 2$	–
9	11	–	–	$L \leq 20$	0, 2, 8
10	5	–	–	$L \leq 200$	0, 2, 3, 8, 9
11	6	–	–	$L = n$	–
12	16	–	$k = n$	–	–
13	7	–	–	$L = k - 1$	8
14	5	$n \leq 100\,000$	–	–	0, 2–6
15	4	$n \leq 150\,000$	–	–	0, 2–6, 14
16	1	–	–	–	0–15

Примеры

стандартный ввод	стандартный вывод
3 3 3 2 3 2 1 1 1	3
4 2 3 1 2 1 2 2 1 2 1	7
6 3 2 1 2 1 2 1 2 2 1 3 3 2 1	23
4 4 2 2 3 4 1 1 2 3 4	9
5 3 5 2 3 1 3 1 1 1 2 2 3	13
20 5 2 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 5 1 2 3 4	448



Замечание

Если в игре участвует один человек, он и есть шулер, поэтому вопросы не нужны.

В первом примере слабая тема каждого игрока — тема 1. Если участников хотя бы двое, ведущий задаёт один вопрос по этой теме. Все честные игроки отвечают неверно, а шулер — верно, поэтому ведущий сразу узнаёт его. Рассматриваются три игры с одним участником, две игры с двумя участниками и одна игра со всеми тремя. Поэтому ответ равен $3 \cdot 0 + 2 \cdot 1 + 1 \cdot 1 = 3$.

Во втором примере игроки 1 и 3 имеют сильную тему 1 и слабую тему 2, а игроки 2 и 4 — наоборот. Для игры с четырьмя участниками ведущий задаёт вопрос по теме 1, а затем по теме 2. Каждый честный игрок ошибётся на вопросе по своей слабой теме. Только шулер ответит верно на оба вопроса. Одного вопроса недостаточно: если шулер — один из двух игроков, для которых выбранная тема сильная, оба этих игрока ответят верно и отличить их друг от друга будет невозможно. Таким образом, $f(1, 4) = 2$.

В игре с двумя участниками достаточно спросить слабую тему одного из них. Если он ответит верно, шулер — он; если неверно, шулер — второй игрок. В игре с тремя участниками выбираем тему, которая является слабой для двух из них. Если кто-то из этих двоих ответит верно, он шулер; если оба ошибутся, шулер — третий игрок. Поэтому три игры с двумя участниками и две игры с тремя участниками имеют сложность 1. Общая сумма равна $3 \cdot 1 + 2 \cdot 1 + 2 = 7$.

В шестом примере рассмотрим игру только с игроками $1, 2, \dots, 10$. Ведущий сначала задаёт два вопроса по теме 1. Для игроков 1 и 6 эта тема сильная, поэтому они оба ответят верно. Для каждого из остальных восьми игроков она слабая или средняя: если такой игрок честный, он хотя бы раз ошибётся. Если кто-то из этих восьми игроков ответит верно на оба вопроса, он и есть шулер. Иначе шулер — один из игроков 1 и 6.

Чтобы различить этих двух игроков, ведущий задаёт ещё один вопрос по теме 2. Для игрока 1 эта тема слабая. Если он ответит верно, он шулер. Если он ошибётся, шулер — игрок 6, найденный методом исключения. Для игрока 6 тема 2 средняя, и это первый вопрос по ней, поэтому он ответит верно даже в том случае, если честный. Всего в худшем случае потребуется 3 вопроса: сначала $L = 2$ вопроса по одной теме, затем ещё один по другой.

Двух вопросов недостаточно. Если дважды спросить одну тему, оба игрока, для которых она сильная, ответят верно. Если шулер — один из них, отличить его от второго будет невозможно. Если спросить две разные темы по одному разу, честные игроки могут ошибиться только на своих слабых темах. Каждая тема слабая ровно для двух из первых десяти игроков, поэтому ни одна из выбранных тем не будет слабой как минимум для шести игроков. Все они ответят верно на оба вопроса. Если шулер — один из них, определить, кто именно, будет невозможно. Следовательно, $f(1, 10) = 3$.

В том же примере рассмотрим игру со всеми 20 участниками. Ведущий задаёт по два вопроса на темы 1 и 2, всего 4 вопроса. У каждого игрока только одна сильная тема, поэтому хотя бы одна из выбранных тем для него слабая или средняя. Если игрок честный, на слабой теме он ошибётся сразу, а на средней — на втором вопросе, поскольку $L = 2$. Таким образом, каждый честный игрок хотя бы раз ошибётся, а шулер ответит верно на все четыре вопроса.

Минимальное число вопросов для этой игры равно 4, то есть $f(1, 20) = 4$.