



A1. Классифайд - 1

Ограничение времени	1 секунда
Ограничение памяти	256 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче на проверку необходимо сдать исходный код программы.

Вася открыл собственный классифайд (доску объявлений). Устроен его классифайд следующим образом: для каждого типа товара продавец с номером i может выставить на продажу только одну единицу товара и заранее указывает минимальную цену S_i , за которую он готов его продать. Каждый покупатель может купить только одну единицу товара, покупатель с номером j указывает максимальную цену B_j , за которую он готов купить товар. Раз в день Вася собирает все заявки и распределяет покупателей и продавцов, которые заключат сделку и по какой цене. При этом сделка между продавцом i и покупателем j может состояться только если $S_i \leq B_j$ по любой цене от S_i до B_j (цену назначает Вася).

Вася получает процент с каждой сделки, поэтому он хотел бы максимизировать суммарную стоимость проданных товаров. Помогите Васе определить эту максимальную стоимость.

Формат ввода

В первой строке задается число наборов тестовых данных T . В этой задаче T всегда равно 1.

В первой строке описания каждого набора записано число N ($1 \leq N \leq 10$) — количество продавцов.

В следующей строке записано N чисел S_i ($1 \leq S_i \leq 100$).

В следующей строке записано число M ($1 \leq M \leq 10$).

В следующей строке записано M чисел B_i ($1 \leq B_i \leq 100$).

Описания наборов отделяются друг от друга пустой строкой.

Формат вывода

Для каждого набора входных данных выведите одно число — максимальную суммарную стоимость проданных товаров.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

Каждый набор тестовых данных оценивается в 5 баллов.

Пример

Ввод

Вывод

1
3
10 100 50

199

Ввод

Вывод

4

9 9 100 99

Язык

C++20 (GCC 14.1)

Набрать здесь

Отправить файл

1

Отправить

A2. Классифайд - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Вася открыл собственный классифайд (доску объявлений). Устроен его классифайд следующим образом: для каждого типа товара продавец с номером i может выставить на продажу только одну единицу товара и заранее указывает минимальную цену S_i , за которую он готов его продать. Каждый покупатель может купить только одну единицу товара, покупатель с номером j указывает максимальную цену B_j , за которую он готов купить товар. Раз в день Вася собирает все заявки и распределяет покупателей и продавцов, которые заключат сделку и по какой цене. При этом сделка между продавцом i и покупателем j может состояться только если $S_i \leq B_j$ по любой цене от S_i до B_j (цену назначает Вася).

Вася получает процент с каждой сделки, поэтому он хотел бы максимизировать суммарную стоимость проданных товаров. Помогите Васе определить эту максимальную стоимость.

Формат ввода

В первой строке задается число наборов тестовых данных T . В этой задаче T равно 10.

В первой строке описания каждого набора записано число N ($1 \leq N \leq 1000$) — количество продавцов.

В следующей строке записано N чисел S_i ($1 \leq S_i \leq 1000$).

В следующей строке записано число M ($1 \leq M \leq 1000$).

В следующей строке записано M чисел B_i ($1 \leq B_i \leq 1000$).

Описания наборов отделяются друг от друга пустой строкой.

Формат вывода

Для каждого набора входных данных выведите одно число — максимальную суммарную стоимость проданных товаров.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).
Каждый набор тестовых данных оценивается в 5 баллов.

Набрать здесь

Отправить файл

1

Отправить

В1. Мяу-мяу! - 1

Ограничение времени	1 секунда
Ограничение памяти	256 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче на проверку необходимо сдать исходный код программы.

Мурмурградск — город котиков! И, как и во всех таких городах, в Мурмурградске у каждого котика есть собственный дом, а у каждого домика — уникальный номер. Котики не любят мочить лапки, поэтому некоторые дома соединены специальными котодорожками, полностью защищенными от воды.

Многолетние исследования показали следующие примечательные черты Мурмурградска:

- 1. Дома и котодорожки представляют собой дерево, где дома — вершины, а котодорожки — ребра.
- 2. Между двумя домами есть котодорожка тогда и только тогда, когда котики в этих домах дружат.

Кошачий мэр решил, что для развития общей сплоченности каким-то котикам нужно будет поменять место жительства. При этом мэру важно, чтобы у всех оставалось свое личное пространство, поэтому никакие два котика не должны жить в одном доме после переезда. Мэр также понял, что, во избежание всеобщего хаоса, надо добавить в план немного стабильности: жители домика номер 1 должны остаться на месте.

Но и у котиков тоже есть свои требования, так как они не любят сильно менять обстановку. Им крайне важно, чтобы после переселения в другой домик они смогли добраться до всех своих друзей, пройдя не более чем одну котодорожку.

У мэра слишком много дел, поэтому за помощью он обратился к вам! Он поставил перед вами следующую задачу: посчитать, сколько существует планов переезда, удовлетворяющих и критериям мэра, и критериям населения.

Так как это число может быть очень большим, необходимо посчитать его по модулю 998244353.

Формат ввода

В первой строке записано одно целое число t — количество наборов входных данных. Далее следуют t наборов входных данных.

Каждый набор данных состоит из нескольких строк. Первая строка набора данных содержит одно целое число n — количество вершин в дереве. Далее идут $n - 1$ строк, каждая содержит два целых числа u и v ($1 \leq u, v \leq n, u \neq v$) — две вершины, которые соединены ребром.

Гарантируется, что заданный граф является деревом, в нем отсутствуют петли и кратные ребра.

Формат вывода

Для каждого набора входных данных выведите в отдельной строке одно целое число — количество подходящих планов переезда по модулю 998244353.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

За каждый тест, для которого был найден правильный ответ, начисляется 1 балл.

Тесты	Дополнительные ограничения			
2-4	t	n	дополнительно	
1 – 20	$t = 1$	$n \leq 9$	—	
21 – 35	$t = 1$	$n \leq 15$	Количество подходящих планов переезда не превосходит 10^5	
36 – 50	$t = 1$	$n \leq 25$	Количество подходящих планов переезда не превосходит 10^5	

Пример

Ввод	Вывод
1	4
6	
5 2	
1 3	
5 4	
1 5	
1 6	

Язык C++20 (GCC 14.1)

Набрать здесь

Отправить файл

1

Отправить

В2. Мяу-мяу! - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Мурмурградск — город котиков! И, как и во всех таких городах, в Мурмурградске у каждого котика есть собственный дом, а у каждого домика — уникальный номер. Котики не любят мочить лапки, поэтому некоторые дома соединены специальными котодорожками, полностью защищенными от воды.

Многолетние исследования показали следующие примечательные черты Мурмурградска:

1. Дома и котодорожки представляют собой дерево, где дома — вершины, а котодорожки — ребра.
2. Между двумя домами есть котодорожка тогда и только тогда, когда котики в этих домах дружат.

Кошачий мэр решил, что для развития общей сплоченности каким-то котикам нужно будет поменять место жительства. При этом мэру важно, чтобы у всех оставалось свое личное пространство, поэтому никакие два котика не должны жить в одном доме после переезда. Мэр также понял, что, во избежание всеобщего хаоса, надо добавить в план немного стабильности: жители домика номер 1 должны остаться на месте.

Но и у котиков тоже есть свои требования, так как они не любят сильно менять обстановку. Им крайне важно, чтобы после переселения в другой домик они смогли добраться до всех своих друзей, пройдя не более чем одну котодорожку.

У мэра слишком много дел, поэтому за помощью он обратился к вам! Он поставил перед вами следующую задачу: посчитать, сколько существует планов переезда, удовлетворяющих и критериям мэра, и критериям населения.

Так как это число может быть очень большим, необходимо посчитать его по модулю 998244353.

Формат ввода

В первой строке записано одно целое число t — количество наборов входных данных. Далее следуют t наборов входных данных.

Каждый набор данных состоит из нескольких строк. Первая строка набора данных содержит одно целое число n — количество вершин в дереве. Далее идут $n - 1$ строк, каждая содержит два целых числа u и v ($1 \leq u, v \leq n, u \neq v$) — две вершины, которые соединены ребром.

Гарантируется, что заданный граф является деревом, в нем отсутствуют петли и кратные ребра.

Формат вывода

Для каждого набора входных данных выведите в отдельной строке одно целое число — количество подходящих планов переезда по модулю 998244353.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).

За каждый набор входных данных, для которого был найден правильный ответ, начисляется 5 баллов.

Тесты	Дополнительные ограничения			
2-4	t	n	дополнительно	
1	$t = 10$	$n \leq 10^5$	—	

Набрать здесь

Отправить файл

1

Отправить

С1. Уникальные подстроки - 1

Ограничение времени	5 секунд
Ограничение памяти	64 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче на проверку необходимо сдать исходный код программы.

Вася программирует так интенсивно, что от его ударов по клавиатуре некоторые клавиши с буквами вывалились и не вставляются обратно — такие клавиши он складывал в особый ящик. Клавиатур Вася сломал немало, поэтому некоторые выпавшие клавиши могут встречаться в нескольких экземплярах.

Проводя весеннюю уборку, Вася наткнулся на ящик с клавишами и подсчитал, сколько раз встречается каждая клавиша. Теперь он хочет составить из этих клавиш одну строку и украсить ей интерьер своего кабинета. Вася считает, что чем больше различных подстрок длины k (последовательностей подряд идущих символов длины k) есть в этой строке, тем сильнее эта строка украсит интерьер. Помогите Васе с украшением.

Формат ввода

В первой строке задается количество наборов входных данных T . В этой задаче T всегда равно 1.

В первой строке каждого описания набора дано два целых числа m и k ($1 \leq m \leq 3, 1 \leq k \leq 13$) — число различных типов клавиш и требуемая длина различных подстрок.

В следующих m строках описываются клавиши. Каждое описание состоит из маленькой английской буквы C_i , написанной на клавише, и числа T_i — количества таких клавиш. Гарантируется, что суммарное количество клавиш не превосходит 16.

Формат вывода

Для каждого набора выведите строку, составленную из клавиш. Чем больше в ней различных подстрок длины k — тем выше будет балл.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

Каждый тестовый набор оценивается максимум в 5 баллов. Оценка за набор вычисляется по формуле $5 \times (\frac{UserAns}{BestAns})^5$, где $BestAns$ — максимальное количество различных подстрок длины k среди решений всех участников и жюри, а $UserAns$ — количество различных подстрок длины k в решении участника.

Пример

Ввод

Вывод

1

aabb

2 2

a 2

b 2

Язык

C++20 (GCC 14.1)

Набрать здесь

Отправить файл

C2. Уникальные подстроки - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Вася программирует так интенсивно, что от его ударов по клавиатуре некоторые клавиши с буквами вывалились и не вставляются обратно — такие клавиши он складывал в особый ящик. Клавиатур Вася сломал немало, поэтому некоторые выпавшие клавиши могут встречаться в нескольких экземплярах.

Проводя весеннюю уборку, Вася наткнулся на ящик с клавишами и подсчитал, сколько раз встречается каждая клавиша. Теперь он хочет составить из этих клавиш одну строку и украсить ей интерьер своего кабинета. Вася считает, что чем больше различных подстрок длины k (последовательностей подряд идущих символов длины k) есть в этой строке, тем сильнее эта строка украсит интерьер. Помогите Васе с украшением.

Формат ввода

В первой строке задается количество наборов входных данных T . В этой задаче T равно 10.

В первой строке каждого описания набора дано два целых числа m и k ($1 \leq m \leq 10, 1 \leq k \leq 13$) — число различных типов клавиш и требуемая длина различных подстрок.

В следующих m строках описываются клавиши. Каждое описание состоит из маленькой английской буквы C_i , написанной на клавише, и числа T_i — количества таких клавиш. Гарантируется, что суммарное количество клавиш не превосходит 10 000.

Формат вывода

Для каждого набора выведите строку, составленную из клавиш. Чем больше в ней различных подстрок длины k — тем выше будет балл.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).

Каждый тестовый набор оценивается максимум в 5 баллов. Оценка за набор вычисляется по формуле $5 \times \left(\frac{UserAns}{BestAns}\right)^5$, где $BestAns$ — максимальное количество различных подстрок длины k среди решений всех участников и жюри, а $UserAns$ — количество различных подстрок длины k в решении участника.

Отправить

D1. Красивые раскраски 2047 - 1

Ограничение времени	4 секунды
Ограничение памяти	256 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче на проверку необходимо сдать исходный код программы.

Дан неориентированный планарный граф на n вершинах и целое положительное число k . Назовем раскраску графа *красивой*, если никакие две соседние его вершины не покрашены в один цвет (для лучшего понимания рекомендуем обратиться к разделу "Примечание"). Найдите количество красивых раскрасок графа в k цветов.

Формат ввода

В первой строке ввода находится единственное целое число t — количество тестовых случаев, которые вам будет необходимо обработать ($1 \leq t \leq 100$).

Первая строка каждого тестового случая содержит целые числа n , m и k — количество вершин и рёбер графа, а также количество доступных цветов. Ограничения на параметры смотри в разделе «Система оценивания».

Следующие m строках содержат по два целых числа u , v , задающих рёбра графа.

Формат вывода

Для каждого тестового случая выведите единственное число — количество красивых раскрасок графа по модулю $10^9 + 7$. Гарантируется, что до взятия по модулю ответ не превосходит $50 \cdot 10^6$.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

Каждый тестовый случай оценивается независимо. Правильный ответ оценивается в 2 балла. Неправильный ответ оценивается в 0 баллов. Балл за задачу равен сумме баллов по всем тестовым случаям.

Ниже приведены значения n , m , k в каждом из тестовых случаев.

тест	n	m	k	тест	n	m	k	тест	n	m	k
1	5	6	5	11	6	7	7	21	9	11	4
2	6	7	5	12	7	8	7	22	10	12	4
3	7	8	5	13	8	10	7	23	4	5	11
4	8	10	5	14	9	11	7	24	5	6	11
5	9	11	5	15	10	12	7	25	6	7	11

тест	n	m	k	тест	n	m	k	тест	n	m	k
6	10	12	5	16	4	5	4				
7	11	13	5	17	5	6	4				
8	12	15	5	18	6	7	4				
9	13	16	5	19	7	8	4				
10	5	6	7	20	8	10	4				

Пример

Ввод

Вывод

11080

7 9 4

1 2

2 3

3 4

4 5

3 6

5 6

3 5

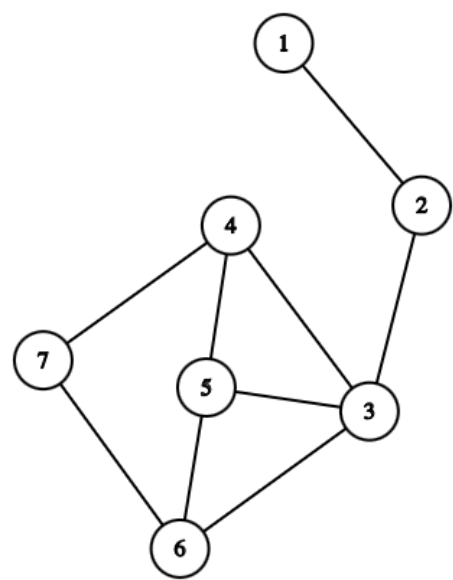
6 7

7 4

Примечания

Планарный граф — граф, который можно изобразить на плоскости без пересечений рёбер не по вершинам.

В тесте из условия задан следующий граф:



Язык

C++20 (GCC 14.1)

Набрать здесь

Отправить файл

D2. Красивые раскраски 2047 - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Дан неориентированный планарный граф на n вершинах и целое положительное число k . Назовем раскраску графа *красивой*, если никакие две соседние его вершины не покрашены в один цвет (для лучшего понимания рекомендуем обратиться к разделу "Примечание"). Найдите количество красивых раскрасок графа в k цветов.

Формат ввода

В первой строке ввода находится единственное целое число t — количество тестовых случаев, которые вам будет необходимо обработать ($1 \leq t \leq 100$).

Первая строка каждого тестового случая содержит целые числа n , m и k — количество вершин и рёбер графа, а также количество доступных цветов. Ограничения на параметры смотри в разделе «Система оценивания». Обратите внимание, что с ростом n значение k в тестах убывает.

Следующие m строк содержат по два целых числа u , v , задающих рёбра графа.

Формат вывода

Для каждого тестового случая выведите единственное число — количество красивых раскрасок графа по модулю $10^9 + 7$.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).

Каждый тестовый случай оценивается независимо. Правильный ответ оценивается в 2 балла. Неправильный ответ оценивается в 0 баллов. Балл за задачу равен сумме баллов по всем тестовым случаям.

Ниже приведены значения n , m , k в каждом из тестовых случаев.

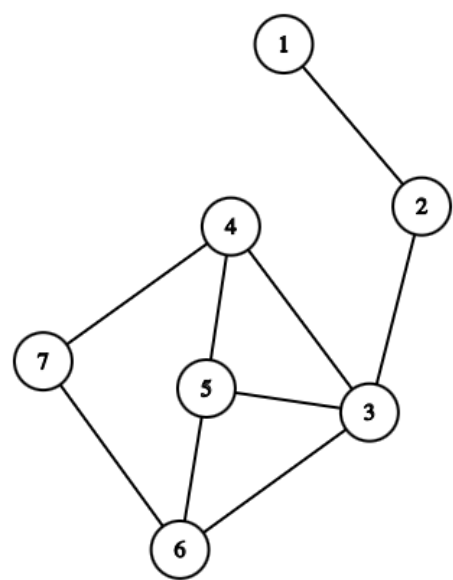
тест	n	m	k	тест	n	m	k	тест	n	m	k
1	24	30	4	11	20	25	5	21	11	13	7
2	22	27	4	12	24	30	4	22	12	15	7
3	20	25	5	13	26	32	4	23	13	16	7
4	20	25	6	14	28	35	4	24	14	17	7
5	19	23	5	15	30	37	4	25	15	18	7
6	18	22	7	16	12	15	5				
7	18	22	8	17	13	16	5				
8	16	20	4	18	14	17	5				

тест	n	m	k	тест	n	m	k	тест	n	m	k
9	16	20	5	19	15	18	5				
10	15	18	4	20	10	12	7				

Примечания

Планарный граф — граф, который можно изобразить на плоскости без пересечений рёбер не по вершинам.

В тесте из условия задан следующий граф:



Набрать здесь

Отправить файл

1

Отправить

E1. Оптимальный опенспейс - 1

Ограничение времени	5 секунд
Ограничение памяти	64 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче на проверку необходимо сдать исходный код программы.

Компания, в которой работает Вася, переехала в новый опенспейс, представляющий собой квадрат размером $S \times S$ и состоящий из ячеек. В каждой из ячеек может расположиться рабочее место сотрудника (сотрудники занумерованы числами от 1 до N) или какое-либо развлечение, например, аэрохоккей или стол с печеньками (развлечений K , они занумерованы отрицательными числами от -1 до $-K$), $N + K = S \times S$.

HR-специалисты компании выяснили важность каждого из развлечений для каждого из сотрудников: кому-то нравится аэрохоккей, а кто-то предпочитает сидеть поближе к печенькам. Некоторым сотрудникам, наоборот, может не нравиться близость к некоторым развлечениям — тогда важность развлечения будет отрицательной. В результате опроса для каждого сотрудника определили K параметров P_1, P_2, \dots, P_K — важность развлечений с номерами $-1, -2, \dots, -K$ соответственно.

Пусть рабочее место сотрудника emp расположено в строке I_{emp} и столбце J_{emp} , а развлечение fun в строке I_{fun} и столбце J_{fun} . Определим близость $dist$, которая будет определяться как $|I_{emp} - I_{fun}| + |J_{emp} - J_{fun}|$ (Манхэттенское расстояние).

Неудовлетворенность сотрудника определяется как сумма произведений близости сотрудника к развлечению на важность этого развлечения для сотрудника, т.е. $unfun = \sum_{i=1}^K dist_i \times P_i$, где $dist_i$ — близость сотрудника к развлечению с номером $-i$, а P_i — важность этого развлечения.

Помогите HR-специалистам определить расположение рабочих мест сотрудников и развлечений так, чтобы минимизировать суммарную неудовлетворенность всех сотрудников.

Формат ввода

В первой строке задается количество наборов входных данных T . В этой задаче T всегда равно 1.

Затем следует T описаний наборов, разделенных пустой строкой.

В первой описания набора строке задается три числа N, K, S ($1 \leq N \leq 8, 1 \leq K \leq 3, 1 \leq S \leq 3, N + K = S \times S$) — количество сотрудников, развлечений и размер опенспейса соответственно.

В следующих N строках описания набора записано по K чисел P_1, P_2, \dots, P_K — важность развлечений для очередного сотрудника.

Формат вывода

Для каждого набора выведите таблицу размером S на S , состоящую из чисел от -1 до $-K$ и чисел от 1 до N — расположение рабочих мест сотрудников и развлечений в опенспейсе.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

Каждый тестовый набор оценивается максимум в 5 баллов. Оценка за набор вычисляется по формуле $5 \times \left(\frac{BestAns}{UserAns}\right)^5$, где $BestAns$ — минимальная неудовлетворенность среди решений всех участников и жюри, а $UserAns$ — неудовлетворенность в решении участника.

Пример

Ввод

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

Вывод

6 -2 1
-1 4 -3
2 3 5

Язык

C++20 (GCC 14.1)

Набрать здесь

Отправить файл

Отправить

Е2. Оптимальный опенспейс - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Компания, в которой работает Вася, переехала в новый опенспейс, представляющий собой квадрат размером $S \times S$ и состоящий из ячеек. В каждой из ячеек может расположиться рабочее место сотрудника (сотрудники занумерованы числами от 1 до N) или какое-либо развлечение, например, аэрохоккей или стол с печеньками (развлечений K , они занумерованы отрицательными числами от -1 до $-K$), $N + K = S \times S$.

HR-специалисты компании выяснили важность каждого из развлечений для каждого из сотрудников: кому-то нравится аэрохоккей, а кто-то предпочитает сидеть поближе к печенькам. Некоторым сотрудникам, наоборот, может не нравиться близость к некоторым развлечениям — тогда важность развлечения будет отрицательной. В результате опроса для каждого сотрудника определили K параметров P_1, P_2, \dots, P_K — важность развлечений с номерами $-1, -2, \dots, -K$ соответственно.

Пусть рабочее место сотрудника emp расположено в строке I_{emp} и столбце J_{emp} , а развлечение fun в строке I_{fun} и столбце J_{fun} . Определим близость $dist$, которая будет определяться как $|I_{emp} - I_{fun}| + |J_{emp} - J_{fun}|$ (Манхэттенское расстояние).

Неудовлетворенность сотрудника определяется как сумма произведений близости сотрудника к развлечению на важность этого развлечения для сотрудника, т.е. $unfun = \sum_{i=1}^K dist_i \times P_i$, где $dist_i$ — близость сотрудника к развлечению с номером $-i$, а P_i — важность этого развлечения.

Помогите HR-специалистам определить расположение рабочих мест сотрудников и развлечений так, чтобы минимизировать суммарную неудовлетворенность всех сотрудников.

Формат ввода

В первой строке задается количество наборов входных данных T . В этой задаче $T = 10$.

Затем следует T описаний наборов, разделенных пустой строкой.

В первой описания набора строке задается три числа N, K, S ($1 \leq N \leq 99, 1 \leq K \leq 10, 1 \leq S \leq 10, N + K = S \times S$) — количество сотрудников, развлечений и размер опенспейса соответственно.

В следующих N строках описания набора записано по K чисел P_1, P_2, \dots, P_K — важность развлечений для очередного сотрудника.

Формат вывода

Для каждого набора выведите таблицу размером S на S , состоящую из чисел от -1 до $-K$ и чисел от 1 до N — расположение рабочих мест сотрудников и развлечений в опенспейсе. Если вы не можете составить решение для какого-либо набора — выведите для него единственное число 0 и этот набор будет пропущен при проверке.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).

Каждый тестовый набор оценивается максимум в 5 баллов. Оценка за набор вычисляется по формуле $5 \times \left(\frac{BestAns}{UserAns}\right)^5$, где $BestAns$ — минимальная неудовлетворенность среди решений всех участников и жюри, а $UserAns$ — неудовлетворенность в решении участников.

Набрать здесь

Отправить файл

1

Отправить

F1. Планетная система - 1

Ограничение времени	2 секунды
Ограничение памяти	256 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче на проверку необходимо сдать исходный код программы.

Давным-давно в далекой галактике существовала планетная система, состоящая из n планет. Каждую планету этой системы можно представить как точку с целыми неотрицательными координатами в m -мерном пространстве, таким образом, каждая планета имеет m координат.

Обозначим за $c_{i,j}$ - j -ю координату i -й планеты.

Известно, что для каждой планеты $i(1 \leq i \leq n)$ выполнено неравенство $0 \leq c_{i,j} < q_j$ по всем $j(1 \leq j \leq m)$. Иными словами, j -я координата любой планеты строго меньше некоторого значения q_j .

Так же известно, что в системе не существует планет с совпадающими координатами.

Находясь на планете a , можно совершить прямой перелет на другую планету b , если и только если каждая координата планеты b не превосходит соответствующей координаты планеты a . Более формально, с планеты a можно совершить прямой перелет на планету b , если для всех j от 1 до m выполнено неравенство: $c_{b,j} \leq c_{a,j}$.

В системе есть k обитаемых планет, имеющих номера x_1, x_2, \dots, x_k , все остальные планеты считаются необитаемыми.

Назовем последовательность планет y_1, y_2, \dots, y_r путем, если для каждого $i(1 \leq i < r)$ можно напрямую перелететь с планеты y_i на планету y_{i+1} . Два пути будем считать различными, если последовательности планет в них отличаются.

Император галактики просит вас для каждой планеты от 1 до n посчитать количество различных путей, начинающихся в этой планете и заканчивающихся в некоторой обитаемой планете по модулю $10^9 + 7$.

Формат ввода

В первой строке вводится число t - количество планетных систем, для которых нужно найти ответ.

Описание каждой планетной системы начинается с трех чисел $n, m, k(1 \leq n \leq 10^5, 1 \leq m \leq 16, 0 \leq k \leq n)$, разделенных пробелом.

В следующей строке содержатся числа q_1, q_2, \dots, q_m - ограничения на координаты планет.

Гарантируется, что во всех тестах данной задачи $\prod_{j=1}^m q_j \leq 10^5$.

Далее следуют n строк, описывающих координаты планет.

i -я из следующих n строк описывает координаты i -й планеты и содержит целые числа $c_{i,1}, c_{i,2}, \dots, c_{i,m}$ ($0 \leq c_{i,j} < q_j$).

Далее следует строка, содержащая k различных чисел - номера обитаемых планет x_1, x_2, \dots, x_k ($1 \leq x_i \leq n$).

Формат вывода

Для каждой из t планетных систем для каждой планеты в системе выведите количество искомых путей по модулю $10^9 + 7$.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится онлайн (после тура баллы за задачу не изменятся).

За каждую планетную систему, для которой был найден правильный ответ, начисляется 2 балла.

В первом тесте 3 планетных системы, в каждой из них не более 20 планет.

Во втором тесте 5 планетных систем, в каждой из них не более 2000 планет.

В третьем тесте 2 планетные системы, в каждой из них $m = 1$.

В четвертом тесте 5 планетных систем, в каждой из них $m = 2$.

В пятом тесте 10 планетных систем, дополнительных ограничений нет.

Пример

Ввод

2
2 2 1
2 2
0 0
0 1
2
3 2 2
2 2
0 0
0 1
1 0
1 2

Вывод

0
1
1
2
1

Примечания

В первой планетной системе две планеты, первая планета имеет координаты $(0, 0)$, вторая планета имеет координаты $(0, 1)$, планета 2 является единственной обитаемой планетой. Первая планета необитаема, и с нее нельзя никуда перелететь, поэтому количество путей, начинающихся в этой планете и заканчивающихся в обитаемой планете, равно 0. Вторая планета обитаема, поэтому путь, состоящий лишь из нее, нам подходит, в то же время путь $2 \rightarrow 1$ оканчивается в необитаемой планете, следовательно, нам не подходит. Таким образом, ответ для второй планеты равен 1.

Во второй планетной системе три планеты, первая имеет координаты $(0, 0)$, вторая планета имеет координаты $(0, 1)$, третья планета имеет координаты $(1, 0)$. Планеты 1 и 2 являются обитаемыми. Из первой планеты перелететь никуда нельзя, но можно остаться на ней и получить путь, состоящий лишь из первой планеты, которая является обитаемой, следовательно, этот путь подходит нам. Для второй планеты существуют пути $2 \rightarrow 1$, так как планета 1 обитаема, а так же путь, состоящий лишь из второй планеты, поэтому ответ для этой планеты 2. Для третьей планеты существует лишь один путь $3 \rightarrow 1$, заканчивающийся в первой планете, которая является обитаемой, следовательно, ответ для третьей планеты 1.

Язык C++20 (GCC 14.1)

[Набрать здесь](#)

[Отправить файл](#)

Отправить

F2. Планетная система - 2

В этой задаче на проверку необходимо сдать текстовый файл с ответом. Входные данные вы можете скачать, нажав на кнопку с изображением стрелки справа-сверху рядом с кнопкой «Объявления жюри».

Давным-давно в далекой галактике существовала планетная система, состоящая из n планет. Каждую планету этой системы можно представить как точку с целыми неотрицательными координатами в m -мерном пространстве, таким образом, каждая планета имеет m координат.

Обозначим за $c_{i,j}$ - j -ю координату i -й планеты.

Известно, что для каждой планеты i ($1 \leq i \leq n$) выполнено неравенство $0 \leq c_{i,j} < q_j$ по всем j ($1 \leq j \leq m$). Иными словами, j -я координата любой планеты строго меньше некоторого значения q_j .

Так же известно, что в системе не существует планет с совпадающими координатами.

Находясь на планете a , можно совершить прямой перелет на другую планету b , если и только если каждая координата планеты b не превосходит соответствующей координаты планеты a . Более формально, с планеты a можно совершить прямой перелет на планету b , если для всех j от 1 до m выполнено неравенство: $c_{b,j} \leq c_{a,j}$.

В системе есть k обитаемых планет, имеющих номера x_1, x_2, \dots, x_k , все остальные планеты считаются необитаемыми.

Назовем последовательность планет y_1, y_2, \dots, y_r путем, если для каждого i ($1 \leq i < r$) можно напрямую перелететь с планеты y_i на планету y_{i+1} . Два пути будем считать различными, если последовательности планет в них отличаются.

Император галактики просит вас для каждой планеты от 1 до n посчитать количество различных путей, начинающихся в этой планете и заканчивающихся в некоторой обитаемой планете по модулю $10^9 + 7$.

Формат ввода

В первой строке вводится число t - количество планетных систем, для которых нужно найти ответ.

Описание каждой планетной системы начинается с трех чисел n, m, k ($1 \leq n \leq 10^5, 1 \leq m \leq 16, 0 \leq k \leq n$), разделенных пробелом.

В следующей строке содержатся числа q_1, q_2, \dots, q_m - ограничения на координаты планет.

Гарантируется, что во всех тестах данной задачи $\prod_{j=1}^m q_j \leq 10^5$.

Далее следуют n строк, описывающих координаты планет.

i -я из следующих n строк описывает координаты i -й планеты и содержит целые числа $c_{i,1}, c_{i,2}, \dots, c_{i,m}$ ($0 \leq c_{i,j} < q_j$)

Далее следует строка, содержащая k различных чисел - номера обитаемых планет x_1, x_2, \dots, x_k ($1 \leq x_i \leq n$).

Формат вывода

Для каждой из t планетных систем для каждой планеты в системе выведите количество искомых путей по модулю $10^9 + 7$.

Система оценивания

Оценка за эту задачу — 50 баллов, тестирование проводится оффлайн (баллы за задачу будут известны после окончания тура).

За каждую планетную систему, для которой был найден правильный ответ, начисляется 2 балла.

Набрать здесь

Отправить файл

1

Отправить