

Задача A. Boxes with treasure

Имя входного файла: `boxes.in`
Имя выходного файла: `boxes.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

На самом деле сражения с рыцарями для драконов также являются своеобразным спортом — «дракорридой». Этот спорт ничуть не более опасен, чем коррида — десяткам историй об успешных битвах рыцарей с драконами, оставшимся в памяти человечества, соответствуют тысячи рассказов разных авторов об успешных дракорридах, ходящих среди драконов.

Для того, чтобы рыцари охотнее вступали с драконами в поединки, драконы держат в пещерах сокровища. Сокровища обычно спрятаны в сундуках так, что даже очень ловкий и сильный рыцарь должен затратить некоторое время на то, чтобы добраться до сокровищ. Один из вариантов — вложенные друг в друга последовательно сундуки. Сундуки представляют собой кубические ящики с целой длиной стороны. Сундук a может быть вложен в сундук b в том и только том случае, если ребро сундука a строго меньше ребра сундука b . В один сундук может быть непосредственно вложено не более одного сундука. Например, в сундуке с ребром 5 может находиться сундук с ребром 2, а в нём — сундук с ребром 1.

У дракона имеется N сундуков, для каждого из которых известна длина ребра. Требуется собрать эти сундуки описанным выше образом так, чтобы количество получившихся комплектов было минимально.

Формат входного файла

В первой строке входного файла задано одно число N — общее количество сундуков ($1 \leq N \leq 10^5$). Далее заданы N целых положительных чисел, не превосходящих 10^9 , по одному числу в каждой строке — длина ребра каждого из ящиков.

Формат выходного файла

Выведите одно число — минимальное количество комплектов, которые получатся, если собрать все сундуки описанным выше образом.

Примеры

<code>boxes.in</code>	<code>boxes.out</code>
4 6 8 8 4	2
3 3 3 3	3
3 2 123456 123457	1

Задача В. Dragon's Quest

Имя входного файла:	<code>quest.in</code>
Имя выходного файла:	<code>quest.out</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

Для поединка с драконом рыцарь может использовать различные наборы амуниции. Например, щит, благословение и меч, или же катану и кимоно, или же BFG-9000. На данный момент запасы амуниции у героя отсутствуют. Рыцарь может заказать любой набор оружия и доспехов у мастеров, при этом отдельные части амуниции изготавливаются параллельно, а время на изготовление каждой части постоянно. В нашем примере благословение он может получить в любой момент, щит делается 2 дня, меч - 4 дня, катана — 8 дней, кимоно - 3 дня, BFG9000 — 100 дней. Таким образом, при первом выборе герой будет экипирован за 4 дня, при втором — за 8, при третьем — за 100.

Дракон, который недоволен длительным отсутствием дракорриды, хочет через подставных лиц подсказать рыцарю такой набор амуниции, который требовал бы для полной экипировки героя наименьшего времени.

Формат входного файла

Входной файл состоит из двух строк. В первой строке перечислены элементы амуниции и время на их изготовление. Каждый элемент задаётся названием (от 1 до 20 строчных латинских букв), после которого через двоеточие следует время на изготовление в днях (от 0 до 1000 включительно). Соседние элементы в списке разделены запятой. Название каждого элемента является уникальным, всего список состоит из не более, чем 20 элементов. Во второй строке заданы наборы амуниции, необходимые и достаточные для битвы рыцаря с драконом. Наборы задаются перечислением названий. Объекты внутри одного набора объединены знаком '&', разные наборы разделены знаком '|'. При этом количество наборов — целое положительное число, не превосходящее 10; также ни один элемент не может входить в один и тот же набор дважды.

Формат выходного файла

Выведите одно число — минимальное количество дней, через которое рыцарь будет экипирован на битву с драконом.

Примеры

<code>quest.in</code>
<code>bless:0,sword:4,shield:2,kimono:3,katana:8,bfg:100</code> <code>bless&sword&shield katana&kimono bfg</code>
<code>quest.out</code>
4

<code>quest.in</code>
<code>gun:13,license:17,vorpalblade:21</code> <code>gun&license license&vorpalblade</code>
<code>quest.out</code>
17

Задача C. Sets of bets

Имя входного файла: `setsofbets.in`
Имя выходного файла: `setsofbets.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Среди драконов популярны ставки на исход поединков дракорриды. Ставки делаются на $m \times n$ ближайших событий сразу. Букмекерская контора выпускает карточки размером $m \times n$, причём каждой из клеток карточки соответствует дракоррида с вероятностью победы дракона p_{ij} . Игрок выбирает n непересекающихся наборов по m клеток так, чтобы в одном наборе никакие две клетки не принадлежали одному и тому же столбцу. Если во всех поединках какого-то набора выигрывает дракон, букмекеры платят игроку 1 драконий доллар.

По заданным вероятностям p_{ij} вычислите математическое ожидание выигрыша с карточки в случае оптимального для игрока распределения клеток по наборам.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m — соответственно число строк и число столбцов в карточке ($1 \leq n \leq 100$, $1 \leq m \leq 10$). Далее задана сама карточка в виде n строк по m чисел в интервале от 0 до 1 включительно — вероятности выигрыша дракона в соответствующем поединке.

Формат выходного файла

Выведите математическое ожидание максимального выигрыша с точностью до 10^{-4} .

Пример

<code>setsofbets.in</code>	<code>setsofbets.out</code>
2 4 1.0 0.2 1.0 0.3 0.5 1.0 0.4 1.0	1.01200

Задача D. Tournament

Имя входного файла: `tournament.in`
 Имя выходного файла: `tournament.out`
 Ограничение по времени: 2 seconds
 Ограничение по памяти: 64 Mebibytes

Отбор лучших рыцарей для дракорриды проводится в рамках рыцарских турниров. Среди N участников рыцарского турнира, состоящего из $N - 1$ боёв по схеме «на выбывание» (каждый бой идёт между двумя ещё не выбывшими участниками, проигравший выбывает), K принадлежат к ордену Меча и Орала. Гроссмейстер ордена, барон О'Бендер, получивший от короля право провести жеребьёвку турнира, планирует составить такую схему турнира, при которой победителем гарантированно будет представитель ордена.

У гроссмейстера есть список пар участников турнира, для которых в случае поединка между ними заведомо известен победитель. Сможет ли гроссмейстер осуществить свой коварный замысел?

Формат входного файла

В первой строке входного файла заданы 3 целых числа N — общее количество участников турнира ($2 \leq N \leq 10^5$), K — количество участников турнира, принадлежащих к ордену Меча и Орала ($1 \leq K \leq N$), и M — количество пар участников турнира, для которых заведомо известен исход боя ($0 \leq M \leq 10^5$). В следующей строке перечислены номера участников турнира, являющихся членами ордена — K попарно различных целых чисел в интервале от 1 до N . Далее заданы M «предопределённых» результатов — пары из двух различных целых чисел в интервале от 1 до K каждое. При этом первым указывается номер победителя.

Формат выходного файла

Выведите 'yes', если гроссмейстер может составить схему турнира так, чтобы обеспечить победу участника из ордена, и 'no' в противном случае.

Пример

<code>tournament.in</code>	<code>tournament.out</code>
4 1 3 1 1 3 1 4 3 2	yes
3 1 1 3 1 2	no

Задача E. Trade

Имя входного файла: `trade.in`
Имя выходного файла: `trade.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Для того, чтобы дракоррида была интересной, оппонентов следует выращивать в соответствующих условиях. Поэтому драконы неплохо разбираются в экономике феодальных государств и часто поддерживают те или иные проекты, направленные на процветание государства.

В королевстве есть N городов. Некоторые из этих N городов соединены дорогами с двусторонним движением так, что из любого города можно проехать в любой другой по сети дорог единственным способом.

Гильдия купцов собирается открыть в двух городах страны крупные ярмарки. При этом во всех городах, расположенных на пути между ними также будут открыты ярмарки — но уже поменьше. Драконы одобрили это намерение и через созданный ими орден Меча и Орала готовы частично профинансировать открытие ярмарок, однако поставили условие: минимизировать максимальное расстояние, которое требуется пройти жителям находящегося в королевстве города для того, чтобы попасть в город, в котором проходит хоть какая-нибудь ярмарка.

Формат входного файла

В первой строке входного файла задано одно целое число N ($2 \leq N \leq 10^5$) — количество городов в стране. Далее заданы $N - 1$ строк, задающих сеть дорог. Каждая из этих строк содержит 3 целых числа: первые два — номера городов, соединённых этой дорогой (города занумерованы числами от 1 до N), третье задаёт длину дороги l ($1 \leq l \leq 10^4$).

Формат выходного файла

В выходной файл выведите одно число — максимальное расстояние, которое в случае, если требование драконов будет удовлетворено, потребуется пройти жителям находящегося в королевстве города для того, чтобы попасть на ярмарку.

Пример

<code>trade.in</code>	<code>trade.out</code>
6 2 1 13 3 1 18 1 4 8 4 5 15 4 6 11	13

Задача F. Treasures

Имя входного файла: `treasures.in`
 Имя выходного файла: `treasures.out`
 Ограничение по времени: 2 seconds
 Ограничение по памяти: 64 Mebibytes

В результате подготовки очередного чемпионата по дракорриде некоему рыцарю досталась карта сокровищ. Карта представляет собой прямоугольник $m \times n$, на котором есть пометки двух типов:

- Цифра k от 0 до 8 — информация о том, что под k клетками, имеющими с этой клеткой общую точку, закопаны сокровища, а в данной клетке сокровищ нет.
- Точка ‘.’ — про эту клетку ничего не известно.

Рыцарь знает, что общее число различных вариантов расположения сокровищ, удовлетворяющих условиям карты, в соседних с «открытыми» (содержащими цифры) клетками не превосходит 10^5 , и что существует как минимум один такой вариант.

Требуется пометить все клетки, в которых заведомо есть сокровища, и все клетки, в которых сокровищ заведомо нет.

Формат входного файла

В первой строке входного файла заданы два целых числа m и n — ширина и длина карты ($1 \leq m \leq 30$, $1 \leq n \leq 16$). В последующих n строках заданы по m символов в соответствии с условиями задачи.

Формат выходного файла

В выходной файл выведите модифицированную карту в следующем формате: поля с цифрами остаются неизменными, в полях, в которых заведомо есть сокровища, точка заменяется на знак ‘*’, в полях, в которых заведомо нет сокровищ, точка заменяется на знак ‘-’.

Пример

<code>treasures.in</code>	<code>treasures.out</code>
6 7	0001-.
0001..	0002*.
0002..	0003*.
0003..	0002*.
0002..	0001-
0001..	000000
000000	000000
000000	