

А. Просто поток

5 секунд, 1024 мегабайта

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Входные данные

В первой строке записано натуральное число N — количество узлов в системе ($2 \leq N \leq 100$). Известно, что источник имеет номер 1, а сток номер N . Во второй строке записано натуральное M ($1 \leq M \leq 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i , где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leq C_i \leq 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Выходные данные

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3} .

входные данные
2
2
1 2 1
2 1 3
выходные данные
4
1
-3

В. Разрез

2 секунды, 1024 мегабайта

Найдите минимальный разрез между вершинами 1 и n в заданном неориентированном графе.

Входные данные

На первой строке входного файла содержится n ($2 \leq n \leq 100$) — число вершин в графе и m ($0 \leq m \leq 400$) — количество ребер. На следующих m строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

Выходные данные

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

входные данные
3 3
1 2 3
1 3 5
3 2 7
выходные данные
2 8
1 2

С. Улиточки

2 секунды, 1024 мегабайта

Две улиточки Маша и Петя сейчас находятся в на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). По соображениям гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Входные данные

В первой строке файла записаны четыре целых числа — n, m, s и t (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика). В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние). Ограничения: $2 \leq n \leq 10^5, 0 \leq m \leq 10^5, s \neq t$.

Выходные данные

Если существует решение, то выведите YES и на двух отдельных строчках сначала последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

входные данные
3 3 1 3
1 2
1 3
2 3
выходные данные
YES
1 3
1 2 3

Дан оргграф, найти два непересекающихся по ребрам пути из s в t , вывести вершины найденных путей.

Д. Великая стена

2 секунды, 1024 мегабайта

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города A и B , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города A в город B .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника $m \times n$. В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

Входные данные

В первой строке входного файла содержатся числа m и n ($1 \leq m, n \leq 50$). Следующие m строк по n символов задают карту страны. Символы обозначают: «#» — гора, «.» — место, пригодное для постройки стены, «-» — место, не пригодное для постройки стены, «A» и «B» — города A и B .

Выходные данные

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены F , которые необходимо построить. Далее нужно вывести карту в том же формате, как во входном файле. Клетки со стеной обозначьте символом «+».

Если невозможно произвести требуемую застройку, то выведите в выходной файл единственное число -1 .

входные данные
5 5 -.-.. A-.-#- .#.#- -.-.- -.-.-B
выходные данные
3 -.-+.. A-.-#- +.#.#- -.-.- -.-.-B

входные данные
1 2 AB
выходные данные
-1

входные данные
2 2 A# #B
выходные данные
0 A# #B

2 секунды, 1024 мегабайта

К 3141 году человеческая цивилизация распространилась по всей галактике. Для перехода от одной звездной системы к другой используются специальные гипертуннели. Чтобы использовать гипертуннель, нужно прилететь в специальное место рядом с исходной звездой используя ваш космический корабль, активировать гипердрайв, пролететь через гипертуннель, выйти рядом со звездой назначения и лететь на нужную вам планету. Весь процесс занимает ровно один день. Небольшой недостаток системы состоит в том, что по каждому туннелю может пролететь только один космический корабль в день.

Вы работаете в транспортном отделе компании «Intergalaxy Business Machines». Сегодня утром ваш начальник дал вам новую задачу. Чтобы запустить чемпионат по программированию IBM, нужно доставить K суперкомпьютеров от Земли, где находится штаб-квартира компании на планету Эйсизм.

Поскольку суперкомпьютеры очень большие, нужно, для перевозки одного нужен целый космический корабль. Вас попросили найти план доставки суперкомпьютеров, который позволит доставить все компьютеры за минимальное число дней. Поскольку IBM является очень мощной корпорацией, можете считать, что каждый раз, когда вам нужен какой-то гипертуннель, он к вашим услугам. Однако вы все равно можно использовать каждый туннель только один раз в день.

Входные данные

Первая строка входного файла содержит N — число звездных систем в галактике, M — число туннелей, K — число суперкомпьютеров для доставки, S — номер солнечной системы, где находится планета Земля и T — номер звездной системы, где находится Планета Эйсизм ($\{2 \leq N \leq 50\}$, $\{1 \leq M \leq 200\}$, $\{1 \leq K \leq 50\}$, $\{1 \leq S, T \leq N\}$, $\{S \neq T\}$)

Следующие M строк содержат по два разных целых числа и описывают туннели. Для каждого туннеля даются номера звездных систем, которые он соединяет. По туннелю можно путешествовать в обоих направлениях, но помните, что каждый день только один корабль может использовать туннель. В частности, два судна не могут одновременно проходить через один и тот же туннель в противоположных направлениях. Ни один туннель не соединяет звезду с самим собой и две звезды связаны не более чем одним туннелем.

Выходные данные

В первой строке выходного файла выведите L — наименьшее число дней, необходимых для доставки K суперкомпьютеров от звездной системы S до звездной системы T с использованием гипертуннелей.

Следующие L строк должны описывать процесс. Каждая строка должна начинаться с C_i — числа кораблей, которые отправляются из одной системы в другую в этот день. Далее должны следовать C_i пар целых чисел, пара A, B означает, что корабль номер A перемещается из текущей звездной системы в звездную систему B .

входные данные
6 7 4 1 6 1 2 2 3 3 5 5 6 1 4 4 6 4 3

выходные данные

```
4
2 3 2 4 4
3 2 4 3 3 4 6
3 1 4 2 6 3 5
2 1 6 3 6
```

Ф. Групповой турнир

2 секунды, 1024 мегабайта

В нашем капиталистическом и меркантильном мире всё решают деньги, и даже спорт не стал исключением. Все команды-участницы уже купили себе нужное количество очков в следующем сезоне, и местной федерации хоккея осталось только распределить результаты предстоящих игр. Однако, некоторые команды не поспешили и помимо покупки очков также купили ещё и результаты некоторых игр. Поначалу в федерации думали, что это им только упростит задачу: чем для большего числа игр результаты уже определены, тем меньше работы остаётся им. Но позже они поняли, что ошиблись. Они попросили вас стать участником их коррупционной схемы и помочь с распределением результатов игр предстоящего сезона.

Местный хоккейный турнир проходит по круговой системе: в турнире участвуют N команд и каждая команда играет с каждой ровно одну игру. За игру команды получают очки по следующим правилам:

- Если победителя удалось выявить в основное время матча, то ему достаётся 3 очка, а проигравшему — 0.
- Если основное время закончилось вничью и для выявления победителя понадобилось дополнительное время (овертайм), то победителю дают 2 очка, а проигравшему — 1 очко. Овертайм не ограничен во времени и длится до тех пор, пока одна из команд не забьёт гол.

По итогам турнира очки команды определяются как сумма её очков по всем сыгранным играм.

Входные данные

В первой строке входного файла содержится целое число N — количество участников турнира ($2 \leq N \leq 100$). Команды занумерованы числами от 1 до N .

Следующие N строк файла содержат по N символов и представляют собой турнирную таблицу на данный момент. Символ a_{ij} в строке i ($1 \leq i \leq N$) на позиции j ($1 \leq j \leq N$) означает результат игры команды номер i с командой номер j и может быть одним из:

- 'W' — означает, что команда i обыгрывает команду j в основное время матча;
- 'w' — команда i обыгрывает команду j в овертайме;
- 'L' — команда i проигрывает команде j в овертайме;
- 'l' — команда i проигрывает команде j в основное время матча;
- '.' — если результат игры между командами i и j ещё не определён;
- '#' — если i равно j , означает отсутствие данного матча, т. к. команда не может играть сама с собой.

Гарантируется, что данная таблица корректна. Более формально:

- $a_{ij} = \text{'\#'}$ для всех $i = j$;
- если $a_{ij} = \text{'.'}$, то $a_{ji} = \text{'.'}$;
- $a_{ij} = \text{'W'}$ тогда и только тогда, когда $a_{ji} = \text{'L'}$;
- $a_{ij} = \text{'w'}$ тогда и только тогда, когда $a_{ji} = \text{'l'}$.

Последняя строка входного файла содержит N целых чисел p_i — количество очков, которое требуется набрать i -й команде ($1 \leq i \leq N$).

Выходные данные

В выходной файл выведите полностью заполненную турнирную таблицу в формате, аналогичном формату входного файла.

Гарантируется, что решение существует. Если решений несколько, то можно вывести любое из них.

входные данные

```
4
#. .W
.#w.
.l#.
L..#
8 6 3 1
```

выходные данные

```
#www
l#wW
LL#w
LLL#
```

Г. Максимальный поток минимальной стоимости

5 секунд, 512 мегабайт

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости из вершины с номером 1 в вершину с номером n .

Входные данные

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 100$, $0 \leq m \leq 2000$). Следующие m строк содержат по четыре целых числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости неотрицательны и не превосходят 10^3 .

Выходные данные

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером n . Ответ не превышает $2^{63} - 1$.

входные данные

```
4 5
1 2 1 2
1 3 2 2
3 2 1 1
2 4 2 1
3 4 2 3
```

выходные данные

```
12
```

Н. Задача о назначениях

2 секунды, 256 мегабайт

Дана целочисленная матрица C размера $n \times n$. Требуется выбрать n ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка, а сумма значений в выбранных ячейках была минимальна.

Входные данные

Первая строка входного файла содержит n ($2 \leq n \leq 300$). Каждая из последующих n строк содержит по n чисел: C_{ij} . Все значения во входном файле неотрицательны и не превосходят 10^6 .

Выходные данные

В первую строку выходного файла выведите одно число — искомая минимизируемая величина. Далее выведите n строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

входные данные
3
3 2 1
1 3 2
2 1 3
выходные данные
3
2 1
3 2
1 3

I. Камень, ножницы, бумага — 2

1 секунда, 512 мегабайт

Год назад Ростислав с Мирославом играли в камень, ножницы, бумагу на щелбаны. За каждый выигранный раунд победитель ставил один щелбан проигравшему. В случае ничьи щелбаны не ставились. Эта игра запомнилась Мирославу как самая худшая игра в его жизни: всю следующую неделю у него болел лоб.

Воспоминания нахлынули на Мирослава, когда он нашел бумажку с шестью числами — запись с той самой игры. Прошло много времени, и теперь Мирослав может спокойно подумать, почему он проиграл так много раз. Но, к сожалению, он не может посчитать точное количество своих поражений, так как он записал только то, что Ростислав показал камень r_1 раз, ножницы s_1 раз и бумагу p_1 раз, а сам Мирослав показал камень r_2 раз, ножницы s_2 раз и бумагу p_2 раз.

Помогите Мирославу узнать по этим данным, какое минимальное количество щелбанов он мог получить в той самой роковой игре.

Для справки, победитель этой игры определяется по следующим правилам:

- Камень побеждает ножницы («камень затупляет или ломает ножницы»);
- Ножницы побеждают бумагу («ножницы разрезают бумагу»);
- Бумага побеждает камень («бумага накрывает камень»).

Если игроки показали одинаковый знак, то засчитывается ничья.

Входные данные

В первой строке входных данных три целых числа r_1, s_1, p_1 . Во второй строке три целых числа r_2, s_2, p_2 .

Все числа неотрицательные и не превышают 10^8 , $r_1 + s_1 + p_1 = r_2 + s_2 + p_2$.

Выходные данные

Выходные данные должны содержать единственное число — минимальное количество щелбанов, которые мог получить Мирослав.

входные данные
3 0 0
0 3 0

выходные данные

3

J. Задача коммивояжеров

2 секунды, 256 мегабайт

Есть n городов. Между городами есть ориентированные дороги, у каждой дороги есть стоимость покупки разрешения на проезд. Мы хотим торговать во всех городах. У нас есть неограниченное кол-во коммивояжеров. Для каждого из них мы должны определить список городов, в которых они будут торговать. Каждый коммивояжер будет объезжать все города из своего списка по циклу (он может по пути заезжать в другие города, но не торговать там). Если два (или более) коммивояжеров будут ездить по одной дороге, то каждому из них мы должны купить разрешение на проезд. Если список у коммивояжера состоит только из одного города, то он либо должен регулярно выезжать из города (тоже по какому-то циклу), либо мы должны купить ему прописку (у каждого города есть цена прописки). Наконец, в любом городе должен торговать только один коммивояжер, иначе предприятием заинтересуется налоговая. Нужно минимизировать издержки.

Входные данные

В первой строке два числа n, m — количество городов и количество дорог ($1 \leq n \leq 256, 0 \leq m \leq n(n-1)$).

Во второй строке n чисел a_i — цена прописки для города номер i ($0 \leq a_i \leq 10^9$).

Затем в m строках описаны дороги. Описание дороги из города u в город v со стоимостью разрешения на проезд c выглядит как $u v cost$ ($1 \leq u, v \leq n, u \neq v, 0 \leq c \leq 10^9$). Гарантируется, что между любой парой городов не более 1 дороги в каждом из направлений.

Выходные данные

Выведите одно число — минимальную сумму издержек.

входные данные

3 3
30 25 30
1 2 3
2 3 5
3 1 10

выходные данные

18

K. В поисках невест

2 секунды, 256 мегабайт

Однажды король Флатландии решил отправить k своих сыновей на поиски невест. Всем известно, что во Флатландии n городов, некоторые из которых соединены дорогами. Король живет в столице, которая имеет номер 1, а город с номером n значит своими невестами.

Итак, король повелел, чтобы каждый из его сыновей добрался по дорогам из города 1 в город n . Поскольку, несмотря на обилие невест в городе n , красивых среди них не так много, сыновья опасаются друг друга. Поэтому они хотят добраться до цели таким образом, чтобы никакие два сына не проходили по одной и той же дороге (даже в разное время). Так как король любит своих сыновей, он хочет, чтобы среднее время сына в пути до города назначения было минимально.

Входные данные

В первой строке входного файла находятся числа n , m и k — количество городов и дорог во Флатландии и сыновей короля, соответственно ($2 \leq n \leq 200$, $1 \leq m \leq 2000$, $1 \leq k \leq 100$). Следующие m строк содержат по три целых положительных числа каждая — города, которые соединяет соответствующая дорога и время, которое требуется для ее прохождения (время не превышает 10^6). По дороге можно перемещаться в любом из двух направлений, два города могут быть соединены несколькими дорогами.

Выходные данные

Если выполнить повеление короля невозможно, выведите на первой строке число -1 . В противном случае выведите на первой строке минимальное возможное среднее время (с точностью 5 знаков после десятичной точки), которое требуется сыновьям, чтобы добраться до города назначения, не менее чем с пятью знаками после десятичной точки. В следующих k строках выведите пути сыновей, сначала число дорог в пути, и затем номера дорог в пути в том порядке, в котором их следует проходить. Дороги нумеруются, начиная с единицы, в том порядке, в котором они заданы во входном файле.

входные данные

```
5 8 2
1 2 1
1 3 1
1 4 3
2 5 5
2 3 1
3 5 1
3 4 1
5 4 1
```

выходные данные

```
3.00000
2 2 6
2 3 8
```