

у2018-4-1. Паросочетания

А. Паросочетание

ограничение по времени на тест: 2 секунды
 ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Двудольным графом называется неориентированный граф (V, E) , $E \subseteq V \times V$ такой, что его множество вершин V можно разбить на два множества A и B , для которых $\forall (e_1, e_2) \in E \ e_1 \in A, e_2 \in B$ и $A \cup B = V, A \cap B = \emptyset$.

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор $S \subseteq E$, что для любых двух рёбер $e_1 = (u_1, v_1), e_2 = (u_2, v_2)$ из S $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

Входные данные

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 250$), где n — число вершин в множестве A , а m — число вершин в B .

Далее следуют n строк с описаниями рёбер — i -я вершина из A описана в $(i + 1)$ -й строке файла. Каждая из этих строк содержит номера вершин из B , соединённых с i -й вершиной A . Гарантируется, что в графе нет кратных рёбер. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Выходные данные

Первая строка выходного файла должна содержать одно целое число l — количество рёбер в максимальном паросочетании. Далее следуют l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы рёбер паросочетания в A и B соответственно.

Пример

входные данные	Скопировать
2 2 1 2 0 2 0	
выходные данные	Скопировать
2 1 1 2 2	

В. Такси

ограничение по времени на тест: 2 секунды
 ограничение по памяти на тест: 256 мегабайт

ввод: taxi.in

вывод: taxi.out

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x -координатой и y -координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d) , равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Входные данные

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js в формате hh:mm (в интервале с 00:00 по 23:59), координаты (a, b) точки

отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Выходные данные

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Примеры

входные данные	Скопировать
<pre>2 08:00 10 11 9 16 08:07 9 16 10 11</pre>	
выходные данные	Скопировать
<pre>1</pre>	

входные данные	Скопировать
<pre>2 08:00 10 11 9 16 08:06 9 16 10 11</pre>	
выходные данные	Скопировать
<pre>2</pre>	

С. Максимальное паросочетание

ограничение по времени на тест: 3.5 секунд
ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

Дан двудольный граф. У каждой вершины графа есть вес. Вес ребра — сумма весов его концов. Вес паросочетания — сумма весов рёбер, входящих в паросочетание. Нужно найти паросочетание максимального веса. Заметим, это паросочетание может содержать сколько угодно рёбер, единственное условие — вес паросочетания должен быть максимальным.

Напомним, что паросочетанием в двудольном графе называется набор рёбер этого графа такой, что никакие два ребра набора не имеют общих вершин.

Входные данные

В первой строке заданы размеры долей n и m ($1 \leq n, m \leq 5\,000$) и количество рёбер e ($0 \leq e \leq 10\,000$). Вторая строка содержит n целых чисел от 0 до 10 000 — веса вершин первой доли. Третья строка содержит m целых чисел от 0 до 10 000 — веса вершин второй доли. Следующие e строк содержат рёбра графа. Каждое ребро описывается парой целых чисел $a_i b_i$, где $1 \leq a_i \leq n$ — номер вершины первой доли и $1 \leq b_i \leq m$ — номер вершины второй доли.

Выходные данные

В первой строке выведите w — максимальный вес паросочетания. Во второй строке выведите k — количество рёбер в паросочетании максимального веса. В следующей строке выведите k различных чисел от 1 до e — номера рёбер в паросочетании. Если максимальных по весу паросочетаний несколько, разрешается вывести одно любое.

Примеры

входные данные	Скопировать
<pre>4 3 3 2 0 9 9 1 0 9 1 2 2 1 1 1</pre>	
выходные данные	Скопировать
<pre>3 2 1 2</pre>	

входные данные	Скопировать
<pre>3 2 4 1 2 3 1 2</pre>	

```
2 1
2 2
3 2
```

выходные данные

Скопировать

```
8
2
2 4
```

D. День рождения

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: birthday.in

вывод: birthday.out

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Входные данные

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leq k \leq 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leq m \leq 150$ и $0 \leq n \leq 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Выходные данные

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых, разделённые одним пробелом. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Числа в каждой из этих двух строк разделяются ровно одним пробелом и выводятся в порядке возрастания. Если максимальных наборов несколько, то выведите любой из них.

Разделяйте вывод для разных наборов входных данных одной пустой строкой.

Примеры

входные данные

Скопировать

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

выходные данные

Скопировать

```
4
2 2
1 2
1 2

4
2 2
1 3
1 2
```

E. НЛО

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: ufo.in

вывод: ufo.out

В маленьком городке M начала действовать служба контроля за незаконными полетами НЛО. Первая задача службы — выяснить, сколько НЛО действует в окрестности города. Агенты службы опросили множество свидетелей и составили список случаев встречи с НЛО, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле было

НЛО. Из данных разведки известна максимальная скорость, с которой может лететь НЛО. Аналитики просят вас узнать, какое минимальное количество НЛО могли наблюдать свидетели.

Входные данные

На первой строке входного файла содержатся целые числа n и v — количество случаев наблюдения и максимальная скорость НЛО ($1 \leq n \leq 100$, $1 \leq v \leq 10\,000$). Следующие n строк содержат описания случаев встречи с НЛО в формате «ЧЧ:ММ x y », где ЧЧ:ММ — время встречи, x и y — координаты места, в котором наблюдался НЛО (для простоты будем считать, что все встречи происходили на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

Выходные данные

Выведите в выходной файл одно число — минимальное возможное количество НЛО.

Примеры

входные данные	Скопировать
<pre>4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1</pre>	
выходные данные	Скопировать
<pre>2</pre>	

входные данные	Скопировать
<pre>1 1 12:00 0 0</pre>	
выходные данные	Скопировать
<pre>1</pre>	

F. Замощение доминошками

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 64 мегабайта

ввод: dominoes.in

вывод: dominoes.out

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Входные данные

Первая строка входного файла содержит 4 целых числа n , m , a , b ($1 \leq n, m \leq 100$, $|a| \leq 1\,000$, $|b| \leq 1\,000$). Каждая из последующих n строк содержит по m символов: символ '.' (точка) обозначает занятую клетку поля, а символ '*' (звёздочка) — свободную.

Выходные данные

В выходной файл выведите одно число — минимальную сумму денег, имея которую, можно замостить свободные клетки поля (и только их).

Пример

входные данные	Скопировать
<pre>2 3 3 2 .*. .*.</pre>	
выходные данные	Скопировать
<pre>5</pre>	