

## Лабораторная работа 2-4. Поток и паросочетания

### А. Просто поток

ограничение по времени на тест: 5 секунд  
 ограничение по памяти на тест: 256 мегабайт  
 ввод: стандартный ввод  
 вывод: стандартный вывод

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 ( $\cdot$ ), 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  $10^{-3}$ .

#### Примеры

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

### В. Разрез

ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт  
 ввод: стандартный ввод  
 вывод: стандартный вывод

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).

#### Примеры

входные данные	Скопировать
3 3 1 2 3 1 3 5 3 2 7	

выходные данные

Скопировать

2 8  
1 2

С. Улиточки

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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). В виду соображений 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: .

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.

Примеры

входные данные

Скопировать

3 3 1 3  
1 2  
1 3  
2 3

выходные данные

Скопировать

YES  
1 3  
1 2 3

Примечание

Дан орграф, найти два непересекающихся по ребрам пути из  $S$  в  $t$ , вывести вершины найденных путей.

D. 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  $(V, E)$ , такой, что его множество вершин  $V$  можно разбить на два множества  $A$  и  $B$ , для которых , и .

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор , 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 ребер. 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

Е. Шахматная доска

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Вася любит играть в необычные шахматы. Его братишка Коля был еще очень маленький. Как-то раз, когда Вася вернулся из школы, он увидел, что его любимую шахматную доску кто-то перекрасил. Вася не сильно разозлился на Колю, потому что очень любил своего брата. Так как у них дома были только черная и белая краски, каждая клетка доски была покрашена в один из этих двух цветов.

Вася решил исправить ошибку брата, он решил покрасить доску так, чтобы она снова стала шахматной. Но Вася почему-то подумал, что хочет красить только диагонали. Причем Вася решил не тратить много времени, поэтому его интересует способ покраски, который содержит наименьшее количество действий. За одно действие Вася может покрасить полностью какую-либо диагональ, в любой из двух цветов: черный или белый. Диагонали бывают двух типов, в зависимости от направления прямой, на которой лежит диагональ. Диагональ, которая лежит на прямой, направленной влево и вниз, является диагональю первого типа, а диагональ, которая лежит на прямой, направленной вправо и вниз, — второго.

Вам предстоит помочь Васе. Задано испорченное Колей шахматное поле. Вам необходимо определить, за какое минимальное количество действий Вася сможет перекрасить доску так, чтобы она стала шахматной.

Входные данные

В первой строке входного файла записаны два целых числа:  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ) — количество строчек и количество столбцов шахматного поля соответственно.

В следующих  $n$  строках записано поле, в каждой строке по  $m$  символов. Каждая строка входного файла описывает одну строку шахматного поля. W соответствует белой клетке, B — черной.

Выходные данные

В выходной файл нужно вывести число  $p$ , количество действий, которое потребуется Васе, чтобы его доска снова стала шахматной. В следующих  $p$  строках описаны действия. Каждое действие описано тремя параметрами: тип диагонали, координаты клетки и цвет. Тип диагонали — это число 1 или 2. Координаты клетки — это два целых числа: строка и столбец одной из клеток, которую покрасили этим действием. Цвет — это символ W или B, белый и черный соответственно.

Примеры

<b>входные данные</b>	Скопировать
3 3 WBB BWB BBW	
<b>выходные данные</b>	Скопировать
1 1 1 3 W	

<b>входные данные</b>	Скопировать
3 3 WBW WWB WWW	
<b>выходные данные</b>	Скопировать
1 2 2 1 B	

<b>входные данные</b>	Скопировать
1 3 WWW	
<b>выходные данные</b>	Скопировать
1 1 1 2 B	

Ф. Двигаем предметы

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

На плоскости расположены  $n$  предметов, их нужно переместить в заданные  $n$  позиций. При этом не важно, какой предмет какую из них

займет. Для каждого предмета известна максимальная скорость, с которой его можно перемещать, при этом перемещать все предметы можно одновременно.

Найдите минимальное время, за которое можно переместить предметы на заданные места.

Входные данные

В первой строке записано число  $n$  ( $1 \leq n \leq 50$ ), в следующих  $n$  строках содержатся описания предметов, в  $i$ -ой из которых, находится три числа координаты  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 1000$ ) и максимальная скорость  $v_i$  ( $1 \leq v_i \leq 10$ )  $i$ -ого предмета соответственно. В следующих  $n$  строках содержатся описания финальных позиций, в  $i$ -ой из которых, заданы координаты  $a_i, b_i$  ( $1 \leq a_i, b_i \leq 1000$ )  $i$ -ой финальной позиции.

Выходные данные

Выведите одно число — минимальное время, за которое можно переместить предметы. Погрешность не более  $10^{-4}$ .

Примеры

<div>входные данные</div> <div>2 0 0 1 0 1 1 1 1 1 0</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>1.0</div>	<div>Скопировать</div>

<div>входные данные</div> <div>2 0 0 1 0 1 1 1 1 2 1</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>2.0</div>	<div>Скопировать</div>

<div>входные данные</div> <div>2 0 0 1 5 0 1 5 12 10 12</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>13.0</div>	<div>Скопировать</div>

<div>входные данные</div> <div>2 0 0 2 5 0 1 5 12 10 12</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>12.0</div>	<div>Скопировать</div>

<div>входные данные</div> <div>4 78 520 5 827 239 5 620 200 7 809 269 7 986 496 754 745 772 375 44 223</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>68.45242650102003</div>	<div>Скопировать</div>

Г. Великая стена

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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.

**Примеры**

<b>входные данные</b>	Скопировать
5 5 --... A-#- .#.#- --..- --.-B	
<b>выходные данные</b>	Скопировать
3 --+.. A-+#- +##.- --..- --.-B	

<b>входные данные</b>	Скопировать
1 2 AB	
<b>выходные данные</b>	Скопировать
-1	

<b>входные данные</b>	Скопировать
2 2 A# #B	
<b>выходные данные</b>	Скопировать
0 A# #B	

**Н. Космические перевозки**

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: bring.in  
вывод: bring.out

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  $B$ .

Пример

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

I. Групповой турнир

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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 поскупились 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 одно 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:

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

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.

Пример

входные данные	Скопировать
<pre>4 #..W .#w. .l#. L..# 8 6 3 1</pre>	
выходные данные	Скопировать
<pre>#wWW l#wW Ll#w LLl#</pre>	

J. Тараканы

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Несколько чистоплотных тараканьих семей хотят поселиться в однокомнатной квартире. Квартира состоит из большой комнаты и кухни, соединённых узким коридором. Тараканов совершенно не интересует коридор сам по себе, однако они хотят иметь доступ и к кухне, и к комнате. Для того чтобы тараканья семья имела такой доступ, ей нужен индивидуальный транспортный путь через коридор. Тараканы — умные насекомые, поэтому они решили составить план коридора и определить, какое максимальное количество транспортных путей можно проложить через коридор.

В тараканьем плане коридор представляется бесконечной в обе стороны полосой шириной  $W$  сантиметров. Тараканы начертили прямоугольную систему координат, ось  $X$  которой параллельна направлению коридора. Хозяева квартиры расставили в коридоре некоторое количество массивных предметов. Каждый предмет является прямоугольником со сторонами, параллельными осям координат, и вершинами с целыми координатами в сантиметрах. Границы коридора задаются уравнениями  $y = 0$  и  $y = W$ . На координатной плоскости тараканы нарисовали квадратную сетку со стороной одной клетки, равной 1 см, начиная от границы коридора.

Тараканы договорились, что каждый транспортный путь является бесконечной в обе стороны цепью квадратных клеток. Цивилизованные тараканы никогда не прыгают, поэтому две подряд идущие клетки в этой цепи должны быть соседними по стороне. Цивилизованные тараканы всегда бегают по полу, поэтому никакая клетка цепи не может пересекаться с хозяйской вещью или выходить за пределы коридора. Тараканы не любят бегать по кругу, поэтому все клетки цепи различны. Тараканы не хотят давки в час пик, поэтому они требуют, чтобы различные транспортные пути не пересекались, то есть не имели общих клеток. Транспортный путь должен соединять кухню и комнату, поэтому разные концы цепи должны уходить в разные стороны коридора.

Ваша задача — определить максимальное количество транспортных путей, которые можно проложить через коридор.

Входные данные

В первой строке входного файла записано два целых числа  $N$  и  $W$ .  $N$  — количество предметов в коридоре,  $W$  — ширина коридора в сантиметрах ( $0 \leq N \leq 5000$ ,  $0 < W \leq 10^9$ ).

Каждая из последующих  $N$  строк описывает одну из хозяйских вещей. Она содержит четыре целых числа  $X_1, Y_1, X_2, Y_2$  — координаты двух противоположных углов прямоугольника ( $-10^9 \leq X_1 < X_2 \leq 10^9, 0 \leq Y_1 < Y_2 \leq W$ ). Предметы, расставленные в коридоре, могут пересекаться.

Выходные данные

В выходной файл нужно вывести одно целое число — максимально возможное количество транспортных путей.

Пример

входные данные	Скопировать
<pre>2 9 -4 4 -1 7 2 1 5 5</pre>	
выходные данные	Скопировать
<pre>5</pre>	

