

Содержание

Must have	2
Задача 2А. Задача о назначениях [0.5 sec, 256 mb]	2
Обязательные задачи	3
Задача 2В. Паросочетание в случайном графе [0.1 sec, 256 mb]	3
Задача 2С. Толстые хоббиты [0.2 sec, 256 mb]	4
Дополнительные задачи	5
Задача 2D. Трисочетание [1 sec, 256 mb]	5
Задача 2Е. Игры на графе [0.5 sec, 256 mb]	6
Задача 2F. Мошенники [0.5 sec, 256 mb]	7

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Must have

Задача 2А. Задача о назначениях [0.5 сек, 256 mb]

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 и сумма значений в выбранных ячейках было минимальна.

Формат входных данных

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 .

Формат выходных данных

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.

Примеры

stdin	stdout
3	3
3 2 1	2 1
1 3 2	3 2
2 1 3	1 3

Подсказка по решению

Крепитесь. Понимание придёт.

Обязательные задачи

Задача 2В. Паросочетание в случайном графе [0.1 sec, 256 mb]

Дан неориентированный граф, нужно выделить в нем максимальное паросочетание.

Гарантируется, что граф сгенерирован следующим методом: фиксируется количество вершин N и количество ребер M , далее говорится, что все концы всех ребер — случайные числа от 1 до N

Соответственно, граф может содержать петли и кратные ребра.

Формат входных данных

N ($1 \leq N \leq 200$) и M ($0 \leq M \leq 1000$).

Формат выходных данных

Выведите на первой строке количество ребер в паросочетание. Далее собственно ребра, каждое на отдельной строке, в произвольном порядке.

Пример

stdin	stdout
5 7	2
1 2	5 4
2 3	1 3
3 1	
3 4	
4 5	
1 2	
5 3	

Замечание

На случайном графе отлично работают простые «Куноподобные» способы.

Задача 2С. Толстые хоббиты [0.2 sec, 256 mb]

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора... В последний поход против Мордора Гэндальф решил отправить N хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуясь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелей другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

Формат входных данных

В первой строке дано целое число N — количество хоббитов ($2 \leq N \leq 100$). Все хоббиты пронумерованы целыми числами от 1 до N . В следующих N строках записана матрица размера $N \times N$. Если i -й и j -й хоббит взвешивались на чашечных весах и оказалось, что i -й хоббит тяжелее, то в i -й строке матрицы на j -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

Пример

stdin	stdout
2	1
0 1	2
0 0	
3	3
0 0 0	1 2 3
0 0 0	
0 0 0	

Замечание

Антицепь.

Дополнительные задачи

Задача 2D. Трисочетание [1 sec, 256 mb]

Дан трёхдольный полный взвешенный граф. Все три доли имеют одинаковый размер n . Трёхдольный означает, что нет ребер внутри доли. Полный означает, что между долями проведены все возможные ребра, каждое ребро ровно один раз. Нужно найти трисочетание минимального веса. Трисочетание — n троек индексов (a_i, b_i, c_i) такие, что все a_i различны, все b_i различны, все c_i различны. Вес трисочетания равен $\sum_{i=1..n} (w[0, a_i, b_i] + w[1, b_i, c_i] + w[2, c_i, a_i])$, где $w[i, a, b]$ — вес ребра из a -й вершины i -й доли в b -ю вершину $(i + 1) \bmod 3$ доли.

Формат входных данных

На первой строке число $n \geq 1$.

Следующие n строк содержат матрицу $w[0]$.

Следующие n строк содержат матрицу $w[1]$.

Следующие n строк содержат матрицу $w[2]$.

Гарантируется, что все веса — случайные числа от k до $2k$ для некоторого k . Все веса — целые числа от 0 до 10^5 .

Формат выходных данных

На первой строке выведите суммарный вес найденного трисочетания. На каждой из следующих $n \leq 13$ строк выведите $a_i b_i c_i$ (вершины нумеруются от 0 до $n - 1$). Тройки можно выводить в любом порядке. Если трисочетаний с минимальным суммарным весом несколько, подойдет любое.

Примеры

stdin	stdout
3	330
64 100 96	0 2 0
90 43 50	1 1 1
12 94 23	2 0 2
75 97 45	
84 19 45	
11 60 28	
9 16 79	
49 21 54	
85 2 74	

Замечание

Это NP-трудная задача.

Более того, даже для $n \leq 13$ непонятно, как быстро решать.

Авторское решение пользуется случайностью входных данных.

Задача 2Е. Игры на графе [0.5 сек, 256 mb]

Коля и Петя любят играть в следующую игру на лекциях по теории сложности. Они рисуют двудольный граф G на листке бумаги и ставят фишку в одну из вершин графа. Далее они ходят по очереди, Коля ходит первым.

Ход состоит из перемещения фишки по ребру в графе. После перемещения фишки вершина, из которой фишка только что ушла, удаляется из графа вместе со всеми инцидентными ей рёбрами. Проигрывает игрок, который не может ходить.

Вам дан граф, который нарисовали Коля и Петя. Для каждой вершины графа определите, кто выиграет, если изначально фишка находится в этой вершине. Предполагайте, что и Коля, и Петя играют оптимально.

Формат входных данных

Первая строка файла содержит три целых числа n_1, n_2, m – число вершин в одной доли, второй доли и число рёбер, соответственно ($1 \leq n_1, n_2 \leq 500, 0 \leq m \leq 50\,000$). Следующие m строк описывают рёбра графа – каждая строка содержит номера вершин, которые рёбро соединяет. Вершины в каждой из долей нумеруются с 1.

Формат выходных данных

Выведите две строке. На первой строке выведите n_1 символов, i -й символ должен быть 'N', если при начале игры из i -й вершины первой доли выиграет Коля, или 'P', если выиграет Петя. Аналогично во второй строке выведите n_2 символов – результаты игры для второй доли графа.

Пример

stdin	stdout
3 3 5	NPP
1 1	NPP
1 2	
1 3	
2 1	
3 1	

Замечание

Да, да, это тоже про паросочетания.

Задача 2F. Мошенники [0.5 sec, 256 mb]

Студенты придумали N тем для курсовых, название каждой состоит из двух слов из заглавных букв английского алфавита. Среди студентов есть читеры, которые вместо того, чтобы придумывать собственные названия, взяли оба слова из уже придуманных их коллегами тем. При этом в качестве первого слова они взяли первое слово одной из тем, а в качестве второго второе. Вам даны все N названий в произвольном порядке, определите максимальное число читеров среди студентов.

Формат входных данных

Первая строка содержит число тестов T . Каждый тест содержит N и N пар слов – названия тем. Все названия различны. $1 \leq T \leq 100$, $1 \leq N \leq 1000$, длины слов от 1 до 20.

Формат выходных данных

Выведите максимальное число читеров. Формат вывода смотрите в примере.

Примеры

stdin	stdout
3	Case #1: 1
3	Case #2: 0
HYDROCARBON COMBUSTION	Case #3: 0
QUAIL BEHAVIOR	
QUAIL COMBUSTION	
3	
CODE JAM	
SPACE JAM	
PEARL JAM	
2	
INTERGALACTIC PLANETARY	
PLANETARY INTERGALACTIC	