

Содержание

07.Base [1/1]	3
Задача A. Pairs. Паросочетание [0.5 sec, 256 mb]	3
07.Advanced [3/5]	4
Задача B. Минимальное контролирующее множество [0.5 sec, 256 mb]	4
Задача C. Такси [0.5 sec, 256 mb]	5
Задача D. День рождения [0.5 sec, 256 mb]	6
Задача E. Замощение доминошками [0.5 sec, 256 mb]	7
Задача F. Испорченный паркет [0.5 sec, 256 mb]	8
07.Hard [0/3]	9
Задача G. Толстые хоббиты [0.5 sec, 256 mb]	9
Задача H. Предприятие переезжает [0.5 sec, 256 mb]	10
Задача I. Сложный вопрос [0.5 sec, 256 mb]	11

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/1152/>

Дедлайн на задачи: 2 недели, до 12-го апреля 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <https://compscicenter.ru/courses/algorithms-2/2015-spring/>

Семинары ведёт Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии указан таймлимит для C/C++.

Таймлимит для Java примерно в 2 раза больше.

Таймлимит для Python примерно в 4 раза больше.

07.Base [1/1]

Задача А. Pairs. Паросочетание [0.5 сек, 256 mb]

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

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

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

Формат входных данных

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

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

Формат выходных данных

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

Пример

pairs.in	pairs.out
2 2	2
1 2 0	1 1
2 0	2 2

07.Advanced [3/5]

Задача В. Минимальное контролирующее множество [0.5 sec, 256 mb]

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Формат входных данных

В первой строке файла даны два числа m и n ($1 \leq m, n \leq 4000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leq K_i \leq n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит 500 000. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \leq L_i \leq n$ — соответствующая i -й вершине первой доли вершина второй доли, или 0, если i -я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S , после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Пример

minimal.in	minimal.out
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача С. Такси [0.5 sec, 256 mb]

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.

Формат входных данных

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

Формат выходных данных

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

Пример

taxi.in	taxi.out
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

Задача D. День рождения [0.5 sec, 256 mb]

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!

Формат входных данных

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.

Формат выходных данных

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.

Пример

birthday.in	birthday.out
2	4
2 2	2 2
1 2 0	1 2
1 2 0	1 2
3 2	
1 2 0	4
2 0	2 2
1 2 0	1 3
	1 2

Задача Е. Замощение доминошками [0.5 сек, 256 mb]

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.

Формат входных данных

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \leq n, m \leq 100$, $|a| \leq 1000$, $|b| \leq 1000$). Каждая из последующих n строк содержит по m символов: символ "." (точка) обозначает занятую клетку поля, а символ "*" (звёздочка) — свободную.

Формат выходных данных

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

Пример

dominoes.in	dominoes.out
2 3 3 2 .** .*.	5

Задача F. Испорченный паркет [0.5 sec, 256 mb]

Пол в некоторой комнате размером $M \times N$ замощен паркетом. При этом некоторые плитки паркета оказались испорчены. Петя решил сделать ремонт в этой комнате, заменив только испорченные клетки. Придя в магазин, он обнаружил, что паркетные плитки бывают двух типов — размера 1×2 , которые стоят A рублей (немного подумав, Петя понял, что плитки 1×2 можно поворачивать на 90 градусов, получая тем самым плитки 2×1) и размера 1×1 , которые стоят B рублей. Разрезать плитку размера 1×2 на две размера 1×1 Петя не может.

Определите, какая минимальная сумма денег нужна Пете, чтобы сделать ремонт.

Формат входных данных

Первая строка входного файла содержит 4 числа N, M, A, B ($1 \leq N, M \leq 300$, A, B — целые числа, по модулю не превосходящие 1000). Каждая из последующих N строк содержит по M символов: символ «.» (точка) обозначает неиспорченную плитку паркета, а символ «*» (звездочка) — испорченную. В конце строк могут идти незначащие пробелы. В конце файла могут быть пустые строки.

Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно заменить испорченные паркетин (и только их).

Пример

floor.in	floor.out
2 3 3 2 .** .*.	5

07.Hard [0/3]

Задача G. Толстые хоббиты [0.5 сек, 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, жалуюсь 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 тяжелее 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

hobbits.in	hobbits.out

Задача Н. Предприятие переезжает [0.5 sec, 256 mb]

На новом суперсовременном предприятии расположено N устройств, каждое из которых соединено проводами с некоторыми из M контактных щитков; каждое устройство соединено с каждым щитком не более чем одним проводом. Предприятие очень автоматизировано: информация о том, что куда подключено, хранится сразу на очень многих компьютерах. Каждый контактный щиток и каждое устройство оборудовано механическим захватом, позволяющим за одну минуту отсоединить провод от щитка или устройства, но на изменение записей потребуется еще пятьдесят девять минут. Отсоединять провода можно, только подавая команды с головного компьютера. Таким образом, после подачи команды следующую команду работы с тем же щитком или с тем же устройством можно подавать не ранее, чем через час. Более того, отсоединять провод надо одновременно с обоих концов, иначе во время, пока провод подсоединен только одним концом, в нем испортится система усиления сигнала, требующая электропитания с обоих концов провода.

Сейчас предприятие переезжает в новое здание, поэтому необходимо отсоединить все устройства от всех контактных щитков, и как можно быстрее. Таким образом, необходимо определить, когда должен отсоединяться какой провод. Напишите программу, которая решит эту задачу.

Формат входных данных

Первая строка входного файла содержит два числа N и M — количество устройств и контактных щитков ($1 \leq M, N \leq 300$). В i -й из следующих N строк содержится информация о том, к каким щиткам подключено i -е устройство: первое число в этой строке — M_i — число проводов, выходящих из i -го устройства, а далее следуют M_i чисел, задающих номера щитков, к которым подключены эти провода.

Формат выходных данных

В выходной файл выведите N строк, в i -ю из которых выведите M_i целых чисел — время в часах, когда нужно начать отсоединение соответствующих проводов.

Начальный момент времени — 0 часов.

Пример

move.in	move.out
2 3	1 0 2
3 1 2 3	0
1 3	

Задача I. Сложный вопрос [0.5 сек, 256 mb]

Дискотеки в школе для программистов проходят довольно оригинально.

Мальчики, пронумерованные натуральными числами от 1 до n , по очереди подходят к девочкам и приглашают на танец одну из еще не танцующих. То есть, сначала партнершу приглашает мальчик с номером 1, после него — мальчик с номером 2 и т.д. Разумеется, мальчик не станет приглашать не симпатичную ему девочку. И может так оказаться, что мальчику будет некого пригласить, соответственно он останется без пары.

Между танцами ребята общаются в небольших компаниях. Когда мальчики оказываются в компании без девочек, они начинают беспокоиться, не могут ли все они остаться без пары на следующий танец. Помогите таким компаниям узнать ответ на их вопрос.

Формат входных данных

Первая строка входного файла содержит 2 целых числа n и k ($2 \leq n, k \leq 50$) — количество мальчиков и количество девочек соответственно. Каждая из следующих n строк содержит описание симпатий i -го мальчика: число m_i и m_i чисел от 1 до k — количество и номера девочек в порядке возрастания, которые нравятся i -му мальчику.

Следующая строка содержит количество запросов q ($1 \leq q \leq 1000$). Каждая из следующих q строк содержит число t_j и t_j чисел от 1 до n — количество и номера мальчиков в порядке возрастания, входящих в одну из описанных в условии компаний.

Формат выходных данных

Для каждого запроса выведите единственную строку, содержащую «Yes», если все мальчики в этой компании могут остаться без пары, и «No» иначе.

Примеры

question.in	question.out
3 3 2 2 3 2 1 2 2 2 3 3 1 1 1 2 1 3	No No Yes
3 3 2 2 3 2 2 3 2 1 2 3 1 1 1 2 1 3	No No No
4 4 4 1 2 3 4 4 1 2 3 4 2 1 2 2 1 2 4 1 2 1 3 1 4 2 3 4	No Yes Yes Yes