

Задача А. Дискретные выходные воздействия (4 балла)

Имя входного файла:	discrete.in
Имя выходного файла:	discrete.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Рассмотрим задачу построения управляющих автоматов с дискретными выходными воздействиями по обучающим примерам, или тестам. i -й ($i = 1..m$) тест в обучающем наборе представляет собой две последовательности длины $len[i]$: последовательность входных событий $in[i] \in \{e_0, e_1\}^{len[i]}$ и последовательность дискретных выходных воздействий $out[i] \in \{a, b, \dots, z\}^{len[i]}$.

Задача управляющего автомата — в ответ на последовательности событий $in[i]$ выдавать последовательности $ans[i]$, в определенном смысле близкие к последовательностям $out[i]$. Близость поведения автомата к поведению, показанному в тестах, выражается функцией приспособленности:

$$f = - \sum_{i=1}^m \frac{1}{len[i]} \sum_{j=1}^{len[i]} [out[i][j] \neq ans[i][j]].$$

Последовательности $ans[i]$ формируются следующим образом. Перед обработкой i -го теста автомат находится в стартовом состоянии. Далее ему последовательно подаются на вход события из последовательности $in[i]$. Значение $ans[i][t]$ равно выходному воздействию, сопоставленному переходу автомата, который выполняется в момент времени времени t .

Каркасом управляющего автомата Мили будем называть частично заданный автомат, переходам которого не сопоставлены выходные воздействия. Необходимо расставить выходные воздействия на переходах заданного каркаса автомата так, чтобы максимизировать значение функции приспособленности f .

Формат входного файла

В первой строке входного файла содержатся числа n и m ($1 \leq n \leq 10$, $1 \leq m \leq 2000$) — число состояний управляющего автомата и число тестов в обучающем наборе.

В следующих n строках последовательно перечисляются состояния каркаса автомата от первого до n -го. Каждая строка имеет вид $t_{i,0} t_{i,1}$, где $t_{i,0}$ и $t_{i,1}$ — номера состояний, в которые осуществляется переход из i -го состояния по событиям e_0 и e_1 соответственно. Первое состояние является стартовым.

Далее, в последующих m строках последовательно перечислены тесты в формате $len[i] in[i] out[i]$, где i — номер теста, $1 \leq len[i] \leq 10^4$.

Формат выходного файла

В выходной файл выведите n строк. Строка i должна иметь формат $z_{i,0} z_{i,1}$, где $z_{i,0}$ и $z_{i,1}$ — оптимальные дискретные выходные воздействия на переходах из состояния i по событиями e_0 и e_1 соответственно. Если ответ не единственный, выведите любой.

Пример

discrete.in	discrete.out
2 2	a c
2 1	b c
1 2	
7 0110001 accbabc	
4 1111 cccc	

Задача В. Непрерывные выходные воздействия (5 баллов)

Имя входного файла:	<code>continuous.in</code>
Имя выходного файла:	<code>continuous.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Рассмотрим задачу построения управляющих автоматов с непрерывными выходными воздействиями по обучающим примерам, или тестам. i -й ($i = 1..m$) тест в обучающем наборе представляет собой две последовательности длины $len[i]$: последовательность входных событий $in[i] \in \{e_0, e_1\}^{len[i]}$ и последовательность непрерывных выходных воздействий $out[i] \in \mathbb{R}^{len[i]}$.

Задача управляющего автомата — в ответ на последовательности событий $in[i]$ выдавать последовательности $ans[i]$, в определенном смысле близкие к последовательностям $out[i]$. Близость поведения автомата к поведению, показанному в тестах, выражается функцией приспособленности:

$$f = - \sum_{i=1}^m \frac{1}{len[i]} \sum_{j=1}^{len[i]} (out[i][j] - ans[i][j])^2.$$

Последовательности $ans[i]$ формируются следующим образом. Перед обработкой i -го теста автомат находится в стартовом состоянии. Далее ему последовательно подаются на вход события из последовательности $in[i]$. Значение $ans[i][1]$ равно выходному воздействию, сопоставленному переходу автомата, который выполняется в момент времени $t = 1$. Далее, если в момент времени $t, t > 1$ выполняется переход с выходным воздействием Δu , то $ans[i][t] = ans[i][t - 1] + \Delta u$.

Каркасом управляющего автомата Мили будем называть частично заданный автомат, переходам которого не сопоставлены выходные воздействия. Необходимо расставить выходные воздействия на переходах заданного каркаса автомата так, чтобы максимизировать значение функции приспособленности f .

Формат входного файла

В первой строке входного файла содержится числа n и m ($1 \leq n \leq 10, 1 \leq m \leq 50$) — число состояний управляющего автомата и число тестов в обучающем наборе.

В следующих n строках последовательно перечисляются состояния каркаса автомата от первого до n -го. Каждая строка имеет вид $t_{i,0} t_{i,1}$, где $t_{i,0}$ и $t_{i,1}$ — номера состояний, в которые осуществляется переход из i -го состояния по событиям e_0 и e_1 соответственно. Первое состояние является стартовым.

Далее, в последующих m строках последовательно перечислены тесты в формате $len[i] in[i] out[i]$, где i — номер теста, $1 \leq len[i] \leq 250$. В последовательностях $out[i]$ вещественные числа записаны через пробел и имеют до шести знаков после запятой.

Формат выходного файла

В выходной файл выведите n строк. Строка i должна иметь формат $z_{i,0} z_{i,1}$, где $z_{i,0}$ и $z_{i,1}$ — вещественные числа, равные оптимальным выходным воздействиям на переходах из состояния i по событиями e_0 и e_1 соответственно. Если от выходного воздействия на некотором переходе значение функции приспособленности не зависит, будем считать, что оптимальное

значение для этого воздействия равно нулю. Ответ засчитывается, если каждое выведенное выходное воздействие отличается от оптимального не более чем на 10^{-6} .

Пример

continuous.in	continuous.out
2 2	0.125835 0
1 2	0 0
1 2	
1 0 0.125032	
1 0 0.126637	

Задача С. Умный муравей (6 баллов)

Имя входного файла: `artificial.in`
Имя выходного файла: `artificial.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задано тороидальное поле, состоящее из клеток и имеющее размер $m \times m$. В части клеток поля расположены яблоки.

По полю может перемещаться муравей. Положения муравья задается клеткой, в которой он находится. Изначально муравей находится в левой верхней клетке поля и смотрит вправо. Муравей может видеть, есть ли яблоко в клетке перед ним, и на каждом шаге должен выполнить ровно одно из трех действий:

- повернуться на 90 градусов налево;
- повернуться на 90 градусов направо;
- переместиться на клетку вперед и съесть яблоко, если оно есть в клетке.

Задача муравья — съесть все яблоки, находящиеся на поле, за k шагов. Требуется построить автомат Мили, под управлением которого муравей решит свою задачу. Автомат должен иметь n состояний. На каждом такте автомат получает на вход события e_0 или e_1 в зависимости от отсутствия или наличия яблока в клетке перед муравьем. Переход автомата может быть помечен одним из трех выходных воздействий: L , R и M , обозначающих поворот налево, поворот направо и перемещение вперед соответственно. Гарантируется, что искомый автомат существует.

Формат входного файла

В первой и единственной строке входного файла содержится единственное число — порядковый номер теста. Все тесты к данной задаче приведены далее в условии.

Формат описания теста таков. В первой строке теста содержатся числа m , n и k . Далее записаны m строк по m символов в каждой, описывающие поле. Звездочка соответствует наличию яблока, точка — отсутствию. Гарантируется, что в левой верхней клетке поля нет яблока.

Формат выходного файла

В i -й строке выходного файла опишите i -е состояние автомата, решающего задачу, в формате $t_{i,0} t_{i,1} a_{i,0} a_{i,1}$, где $t_{i,0}$ и $t_{i,1}$ — номера состояний на переходах по e_0 и e_1 (состояния нумеруются от 1 до n), $a_{i,0}$ и $a_{i,1}$ — выходные воздействия на этих переходах.

Пример

	<code>artificial.in</code>	<code>artificial.out</code>
1		2 1 R M 3 3 M M 3 4 R M 2 2 L M

Номер теста	Тест
1	10 4 22 .*..... .*..... .*****.....*.....*.....*.....*..... .*****.....
2	10 1 117 .***** ***** ***** ***** ***** ***** ***** ***** ***** *****
3	10 3 117 *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.* *.*.*.*.*
4	10 4 22 .**.....*.....*.....**..**

Номер теста	Тест
5	10 3 47 .**.....*.....*.....**..**
6	10 5 48 .***.....*.....*...** *..*...*.. *.*..... *.*...*.. *.*...*.. *..*...*.. *..... .*****.*..
7	10 4 52 .***.....*.....*...** *..*...*.. *.*..... *.*...*.. *.*...*.. *..*...*.. *..... .*****.*..
8	10 5 58 .***.....*.....*...** *..*...*.. *.*..... *.*...*.. *.*...*.. *..*...*.. *..... .*****.*..

Задача D. Интерактивная минимизация (6 баллов)

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

На единичном гиперкубе $[0, 1]^n$ определена функция f , которая задается формулой:

$$f(x_1, \dots, x_n) = \sum_{i=1}^n q_i (x_i - a_i) + q_0 \sum_{j=1}^n \cos(2\pi t_j (x_j - a_j)),$$

где $q_0, \dots, q_n \in [-1, 1]$, $a_1, \dots, a_n \in [0, 1]$, $t_1, \dots, t_n \in [1, 10]$. Требуется найти глобальный минимум функции f . Значения параметров q_0, \dots, q_n , a_1, \dots, a_n и t_1, \dots, t_n неизвестны: чтобы узнать значение $f(x_1, \dots, x_n)$, необходимо сделать запрос проверяющей программе.

Интерактивный протокол

Ваша программа может делать два типа запросов.

Во-первых, программа может спросить значение f в точке (x_1, \dots, x_n) . После этого программа может прочитать ответ на запрос из стандартного входа. Число запросов данного типа не должно превышать $1000n$.

Во-вторых, программа может дать ответ на задачу, указав минимум функции f . После запроса второго типа программа должна прекратить свое взаимодействие с проверяющей программой.

В конце каждого запроса необходимо вывести перевод строки и очистить буфер потока стандартного выхода командой `flush`.

Формат входного файла

Первая строка стандартного входа содержит число n ($1 \leq n \leq 20$) — размерность области определения функции f . Последующие строки стандартного входа содержат ответы на запросы вашей программы. Каждая строка входа состоит из единственного вещественного числа — значения f в желаемой точке. Все значения выводятся с девятью знаками после запятой.

Формат выходного файла

При запросе значения функции f строка выходного потока должна состоять из n вещественных чисел от 0 до 1, разделенных пробелом. При запросе с указанием минимума следует вывести слово «`minimum`», после чего через пробел записать ответ на задачу — одно вещественное число. Ответ будет засчитан, если он отличается от глобального минимума функции не более чем на 10^{-3} .

Пример

standard input	standard output
2	0.25 0.25
0.120000000	0.35 0.35
0.110000000	0.45 0.45
0.090000000	0.55 0.55
0.080000000	minimum 0.08

Приведенный пример лишь показывает протокол в действии. При таком небольшом числе запросов трудно определить минимум функции.