

Задача А. Мутация (2 балла)

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

Рассмотрим следующий оператор мутации. Пусть $s \in \{0, 1\}^n$ — битовая строка. В результате мутации каждый бит строки s инвертируется с вероятностью $1/n$. Для заданных пар строк s_i и t_i , $i = 1..m$ найдите вероятность получения t_i при применении рассмотренного оператора мутации к s_i .

Формат входного файла

В первой строке входного файла содержатся числа n и m ($1 \leq n \leq 10$, $1 \leq m \leq 100000$) — размер битовых строк и число пар строк, для которых нужно ответить на вопрос задачи. В каждой последующих двух строках входного файла записаны битовые строки s_i и t_i , $i = 1..m$ в виде последовательностей чисел 0 и 1 длины n .

Формат выходного файла

Для каждой пары строк s_i и t_i выведите в выходной файл строку, состоящую из одного вещественного числа — вероятности получения строки t_i из строки s_i в результате мутации. Ответ будет засчитан, если он отличается от правильного не более, чем на 10^{-9} .

Пример

| <code>mutation.in</code> | <code>mutation.out</code> |
|--------------------------|---------------------------|
| 5 2 | 0.00032 |
| 11111 | 0.00128 |
| 00000 | |
| 11110 | |
| 00000 | |

Задача В. Кроссовер (3 балла)

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

Пусть A — множество, состоящее из m битовых строк длины n , представляющих собой поколение генетического алгоритма. Необходимо для заданной битовой строки s (также размера n) выяснить, может ли она получиться из каких-либо двух строк из множества A в результате применения операторов одноточечного, двухточечного и однородного кроссовера.

Отметим, что при выполнении одноточечного и двухточечного кроссоверов точки скрещивания могут находиться как до первого символа строки, так и после последнего символа, а в случае двухточечного кроссовера точки могут совпадать.

Формат входного файла

В первой строке входного файла содержатся два числа m и n ($1 \leq m \leq 300$, $1 \leq n \leq 50$) — число битовых строк в множестве A и их длина. В последующих m строках входного файла записаны последовательности длины n из нулей и единиц, задающие множество A . В последней строке входного файла записана битовая строка s в том же формате, что и строки из множества A .

Формат выходного файла

В первой строке выходного файла должно содержаться слово «YES», если строка s может получиться в результате применения одноточечного кроссовера к каким-либо двум (возможно, одинаковым) строкам из множества A , и слово «NO» в противном случае. Аналогично во второй и третьей строках выходного файла должны содержаться слова «YES» или «NO» в зависимости от того, может ли получиться строка s в результате двухточечного и однородного кроссовера соответственно.

Пример

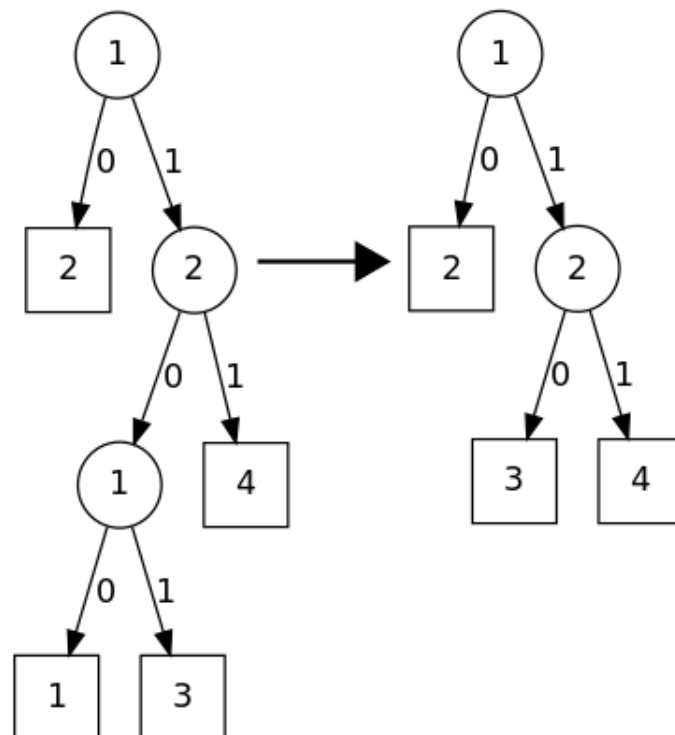
| <code>crossover.in</code> | <code>crossover.out</code> |
|---------------------------|----------------------------|
| 2 4 | NO |
| 0000 | NO |
| 1111 | YES |
| 0101 | |

Задача С. Деревья решений (4 балла)

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

Деревья решений — один из способов задания функции нескольких аргументов, каждый из которых пробегает конечное множество значений. В частности, деревья решений могут использоваться для представления функции переходов автомата. Рассмотрим дерево решений, соответствующее функции переходов некоторого состояния автомата. Будем считать, что переходы из состояния в состояние осуществляются на основе значений m предикатов. В этом случае каждый лист дерева решений помечен значением функции переходов, а каждая вершина ветвления — номером предиката, в зависимости от значения которого при вычислении функции переходов будет выбрана одна из дочерних вершин.

В случае, если построение деревьев решений осуществляется на основе эволюционных вычислений, возникает задача удаления недостижимых ветвей дерева. Вам предстоит решить эту задачу. Более формально, если вершина ветвления v помечена предикатом с номером p и если среди ее предков есть другая вершина u , помеченная предикатом с тем же номером p , то вершина v должна быть удалена вместе с ее недостижимым поддеревом. Оставшееся поддерево вершины v должно быть присоединено к бывшему родителю v . Пример указанного преобразования приведен на рисунке.



Формат входного файла

В первой строке входного файла содержится число k ($1 \leq k \leq 2 \cdot 10^5$) — число вершин дерева решений. В последующих k строках входного файла последовательно с первой по k -ю описаны вершины дерева. Формат задания листа: leaf a_i , где a_i ($1 \leq a_i \leq 10^9$) — значение

функции переходов для i -й вершины. Формат задания вершины ветвления: `choice p_i f_i t_i` , где p_i ($1 \leq p_i \leq 10^9$) — номер предиката для i -й вершины, f_i и t_i — номера дочерних вершин, соответствующих переходам по 0 и 1. Корень дерева является вершиной с номером 1.

Формат выходного файла

Запишите в выходной файл дерево решений, получившееся из исходного в результате удаления всех недостижимых ветвей. Используйте тот же формат, что и во входном файле.

Пример

| trees.in | trees.out |
|---|---|
| 7 choice 1 2 3 leaf 2 choice 2 4 5 choice 1 6 7 leaf 4 leaf 1 leaf 3 | 5 choice 1 2 3 leaf 2 choice 2 4 5 leaf 3 leaf 4 |

Приведенный пример соответствует рисунку.

Задача D. Стартовое состояние (3 балла)

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

Задан управляющий автомат Мура. Пусть множество входных событий для него равно $\{0, 1\}$, а множество выходных воздействий — $\{a, b, \dots, z\}$. При выполнении перехода автомат вырабатывает выходное воздействие, которым помечено состояние, куда был совершен переход.

Задана также последовательность выходных воздействий $z = a_1 \dots a_m$ длины m . Известно, что последовательность z была получена в результате работы автомата на некоторых входных данных. Требуется найти номера состояний, из которых существует последовательность переходов, результатом выполнения которой будет выработка последовательности z .

Формат входного файла

В первой строке входного файла содержатся два числа m и n ($1 \leq m \leq 5 \cdot 10^5$, $1 \leq n \leq 500$) — длина последовательности z и число состояний автомата. В следующих n строках последовательно от первого до n -го описаны состояния автомата Мура. Каждая строка имеет формат $t_{i,0} \ t_{i,1} \ b_i$, где $t_{i,0}$ и $t_{i,1}$ — номера состояний, куда ведут переходы из состояния i по событиям 0 и 1 соответственно, b_i — выходное воздействие, которым помечено состояние i . В последней строке входного файла записана последовательность выходных воздействий z .

Формат выходного файла

В единственной строке выходного файла выведите число состояний, являющихся решением задачи, а затем через пробел перечислите в порядке возрастания номера этих состояний.

Пример

| start.in | start.out |
|-----------------------------|-----------|
| 2 2 2 2 a 2 1 b ab | 1 2 |