

Задача А. Генетические алгоритмы

Имя входного файла: `genetic.in`
Имя выходного файла: `genetic.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Генетические алгоритмы — один из современных методов искусственного интеллекта. Они основаны на принципе естественного отбора, который был предложен Чарльзом Дарвином, и являются одной из разновидностей эволюционных вычислений.

В процессе работы генетического алгоритма поддерживается некоторое множество допустимых решений задачи, называемое *поколением*. Работа алгоритма состоит в генерации очередного поколения. Для этого используются так называемые операции «мутации» и «скрещивания». Как правило, операция мутации состоит в небольшом изменении решения, а операция скрещивания — в построении из двух решений одного или двух новых.

Одним из применений генетических алгоритмов является решение задачи коммивояжера. Допустимыми решениями в этой задаче являются перестановки чисел от 1 до n — такие последовательности длины n , в которых каждое из этих встречается ровно один раз.

Некоторые генетические алгоритмы, решающие задачи коммивояжера, используют операции скрещивания, которые генерируют только так называемые согласованные перестановки. Пусть заданы две перестановки. Назовем третью перестановку *согласованной* с первыми двумя, если для любой пары чисел (i, j) такой, что в обеих перестановках число i идет раньше, чем число j , и в новой перестановке i идет раньше j .

Для анализа таких алгоритмов полезно уметь определять число перестановок, согласованных с данными двумя. Ваша задача состоит в том, чтобы написать программу, которая будет вычислять это число.

Формат входного файла

Первая строка входного файла содержит число n ($1 \leq n \leq 20$). Вторая строка входного файла содержит n чисел — первую перестановку, а третья строка содержит вторую перестановку.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

<code>genetic.in</code>	<code>genetic.out</code>
2 1 2 2 1	2
5 1 4 2 5 3 4 3 1 5 2	14

В первом примере нет ни одной пары чисел (i, j) такой, что в обеих перестановках i идет раньше, чем j . Поэтому с ними согласованы обе возможные перестановки из двух чисел.

Задача В. Новая последовательность

Имя входного файла: `newseq.in`
Имя выходного файла: `newseq.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Паша очень любит искать во всем закономерности. Например, недавно он обнаружил, что в его любимой стратегии «Герои ушей и хвоста» на одном из подуровней количество кроликов от года к году изменяется совсем не случайным образом.

Паша целый вечер бился над тем, что же это за последовательность, но ни одна из ранее им изученных не подошла. На следующий день он вплотную занялся изучением этой совершенно новой для него последовательности. Ближе к обеду у него появилась следующая гипотеза: если в какой-то год кроликов было n , то в следующем году их станет

$$f(n) = \prod_{p|n} (p+1)^{k(n,p)}$$

Здесь p — простое число, а $k(n,p)$ — максимальная степень p , на которую делится n ($p^{k(n,p)}$ делит n , а $p^{k(n,p)+1}$ не делит).

Паше очень нравится эта последовательность, и он теперь хочет узнать, сколько кроликов будет через k лет, если сейчас у него n кроликов.

Например, если $n = 5$, то на следующий год количество кроликов возрастет до шести, на следующий год их будет 12, потом 36, затем 144 и т.д.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и k ($2 \leq n \leq 10^9$, $1 \leq k \leq 5000$).

Формат выходного файла

Поскольку ответ может быть очень большим, выведите в выходной файл его разложение на простые множители. Первая строка выходного файла должна содержать целое число m — количество простых делителей числа. Каждая из следующих m строк должна содержать пару натуральных чисел p_i и k_i , где p_i — простое, а k_i — максимальное число такое, что $p_i^{k_i}$ делит ответ на задачу. Все p_i должны быть различны.

Примеры

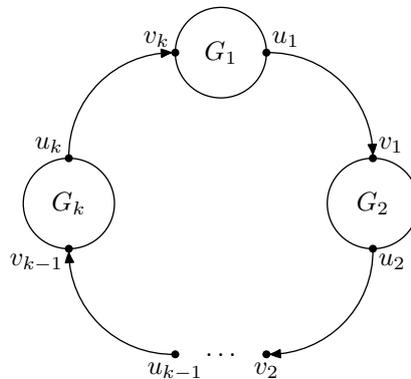
<code>newseq.in</code>	<code>newseq.out</code>
5 1	2 2 1 3 1
5 2	2 2 2 3 1

Задача С. Колеса

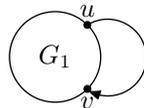
Имя входного файла:	wheels.in
Имя выходного файла:	wheels.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Некоторые результаты в теоретической информатике становятся широко известными, например топологическая сортировка, алгоритм Дейкстры, алгоритм поиска максимального потока. А другие идеи иногда забываются, и остаются только в пыльных журналах на библиотечных полках. В этой задаче мы рассмотрим один алгоритм, придуманный Дональдом Кнудом, который автор называет «декомпозицией графа на колеса».

Вам задан ориентированный сильно связный граф G . Кнут доказал, что каждый такой граф может быть представлен в виде «колеса» — цикла, состоящего из сильно связных подграфов G_1, G_2, \dots, G_k , причем из каждого графа в следующий ведет ровно одно ребро, и других ребер между графами нет.



При этом может случиться, что у цикла всего одно ребро:



Каждый граф G_i , который содержит хотя бы одно ребро, в свою очередь может быть представлен в виде декомпозиции на колеса. Этот процесс продолжается, пока каждая из компонент графа не будет представлять собой одну вершину без ребер.

Для заданного графа найдите его декомпозицию на колеса.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m — количество вершин и ребер в графе, соответственно ($1 \leq n \leq 200$, $1 \leq m \leq 300$). Следующие m строк описывают ребра, каждое ребро описывается парой вершин: началом и концом. Граф может содержать петли, или ребра, направленные навстречу друг другу, но не содержит параллельных ребер. Граф сильно связан.

Формат выходного файла

Выведите граф, описанные следующим образом. Если он содержит одну вершину и не содержит ребер, выведите просто “vertex i ”, где i — номер вершины. В противном случае выведите “wheel k ”, где k — количество компонент вдоль цикла внешнего «колеса». Затем должно следовать k описаний элементов цикла. После каждой части выводите “edge e ”, эта запись описывает ребро из очередного элемента цикла в следующий.

Каждая вершина и каждое ребро исходного графа должны встречаться ровно один раз в описании декомпозиции графа.

Разделяйте выводимые фрагменты пробелами и/или переводами строк. Любое количество пробелов и переводов строк проверяющая программа считает эквивалентным одному пробелу. Вам не требуется форматировать вывод с помощью отступов, как в примерах.

Пример

wheels.in	wheels.out
7 9 1 6 6 2 2 1 5 7 2 3 3 5 4 3 5 4 7 6	wheel 3 wheel 3 vertex 1 edge 1 vertex 6 edge 2 vertex 2 edge 3 edge 5 wheel 3 vertex 3 edge 6 vertex 5 edge 8 vertex 4 edge 7 edge 4 vertex 7 edge 9
3 3 1 2 2 3 3 1	wheel 3 vertex 1 edge 1 vertex 2 edge 2 vertex 3 edge 3
4 5 1 2 2 3 3 4 4 1 1 3	wheel 1 wheel 4 vertex 1 edge 1 vertex 2 edge 2 vertex 3 edge 3 vertex 4 edge 4 edge 5

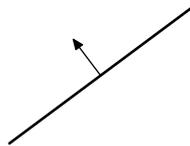
Задача D. Выпуклая оболочка

Имя входного файла: `convex.in`
Имя выходного файла: `convex.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Выпуклой оболочкой множества точек на плоскости называется минимальный выпуклый многоугольник, содержащий все точки из заданного множества. В этой задаче вам надо будет найти среднее значение площади выпуклой оболочки некоторого множества точек за некоторый период.

Дадим некоторые пояснения, что означает «среднее значение». Пусть некоторая величина y изменяется в течение времени t от 0 до T по закону $y = f(t)$. Изобразим график функции $f(t)$ на участке от 0 до T и найдем площадь S под графиком. Тогда среднее значение величины y на указанном промежутке равно S/T .

Вам задано несколько прямых на плоскости. Каждая прямая движется с некоторой постоянной скоростью v_i перпендикулярно самой себе.



Рассмотрим множество их точек пересечения (если две прямые параллельны или совпадают, то считается, что у них нет точек пересечения). По мере того как прямые перемещаются, выпуклая оболочка их точек пересечения меняется. Найдите среднее значение площади их выпуклой оболочки за период от 0 до T .

Формат входного файла

Первая строка входного файла содержит n — количество прямых ($3 \leq n \leq 10$). Следующие n строк содержат описание заданных прямых. Каждая прямая описывается четырьмя целыми числами x_1, y_1, x_2 и y_2 , а также вещественным числом v ($-20 \leq x_1, y_1, x_2, y_2, v \leq 20$, $x_1 \neq x_2$ или $y_1 \neq y_2$, $|v| \geq 0.01$). Прямая, которая задана таким образом, проходит в момент 0 через точки (x_1, y_1) и (x_2, y_2) и перемещается с постоянной скоростью $|v|$ в направлении, перпендикулярном себе. При этом если встать в точку (x_1, y_1) и посмотреть на точку (x_2, y_2) , то прямая перемещается вправо, если v положительна, либо влево, если v отрицательна.

Мы считаем, что ось Oy направлена влево, если встать в начало координат и посмотреть в направлении оси Ox .

Последняя строка входного файла содержит вещественное число T ($0.01 \leq T \leq 100$).

Всегда найдутся три прямые, которые попарно не параллельны друг другу. Никакие две прямые не совпадают в течение ненулевого промежутка времени.

Формат выходного файла

Выведите одно вещественное число — среднюю площадь выпуклой оболочки точек пересечения заданных прямых за период от 0 до T . Ваш ответ должен иметь абсолютную погрешность не более 10^{-4} или относительную погрешность не более 10^{-9} .

Примеры

convex.in	convex.out
4 0 0 0 1 1.0 1 0 1 1 -1.0 0 0 1 0 -1.0 0 1 1 1 1.0 0.2	0.6533333333333333
4 0 0 1 0 0.1 0 0 0 1 0.1 1 0 1 1 0.1 2 0 0 2 -0.3 3.0	1.0237828009

В первом примере выпуклая оболочка — это квадрат со стороной $1 - 2t$. Ее площадь равна $(1 - 2t)^2$, среднее значение площади за заданный период равно $\frac{49}{75}$.