

Задача 1. Топологическая сортировка

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Имеется N лекционных тем, пронумерованных числами от 1 до N . За одну лекцию можно целиком рассказать одну тему, смешивать темы на лекции нельзя. Некоторые темы зависят от других тем. Если тема A зависит от темы B , то это означает, что нельзя рассказывать тему A , не рассказав предварительно тему B . Требуется определить, в каком порядке нужно рассказывать темы на лекции, чтобы соблюсти все зависимости.

Если есть несколько подходящих порядков, нужно выбрать лексикографически наименьший из них. То есть номер первой рассказываемой темы должен быть минимально возможным, среди таких порядков нужно выбрать такой, у которого номер темы на второй лекции минимально возможный, далее нужно минимизировать номер темы на третьей лекции, и так далее.

Указание: Используйте алгоритм, представленный в презентации, который носит имя Кана. Возможно, он был рассказан на дискретной математике.

Формат входных данных

В первой строке записано два целых числа: N — количество тем и M — количество зависимостей ($1 \leq N \leq 400$, $1 \leq M \leq \frac{1}{2}N(N-1)$).

В оставшихся M строках описаны зависимости. Каждая зависимость описывается двумя целыми числами B и A , что означает, что тема A зависит от темы B ($1 \leq A \neq B \leq N$).

Формат выходных данных

Выведите искомый порядок в единственную строку выходного файла: N целых чисел, обозначающих номера тем. Если в темах имеется циклическая зависимость и искомого порядка не существует, выведите слова “bad course” вместо порядка.

Пример

input.txt	output.txt
5 7 1 5 1 3 5 2 5 4 3 4 1 2 4 2	1 3 5 4 2
3 3 1 2 2 3 3 2	bad course

Задача 2. Топологическая сортировка+

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Имеется N лекционных тем, пронумерованных числами от 1 до N . За одну лекцию можно целиком рассказать одну тему, смешивать темы на лекции нельзя. Некоторые темы зависят от других тем. Если тема A зависит от темы B , то это означает, что нельзя рассказывать тему A , не рассказав предварительно тему B . Требуется определить, в каком порядке нужно рассказывать темы на лекции, чтобы соблюсти все зависимости.

Если есть несколько подходящих порядков, нужно выбрать лексикографически наименьший из них. То есть номер первой рассказываемой темы должен быть минимально возможным, среди таких порядков нужно выбрать такой, у которого номер темы на второй лекции минимально возможный, далее нужно минимизировать номер темы на третьей лекции, и так далее.

Указание: Чтобы ускорить алгоритм Кана, для каждой темы храните и поддерживайте количество ещё не рассказанных тем, от которых она зависит.

Формат входных данных

В первой строке записано два целых числа: N — количество тем и M — количество зависимостей ($1 \leq N \leq 5\,000$, $1 \leq M \leq 100\,000$).

В оставшихся M строках описаны зависимости. Каждая зависимость описывается двумя целыми числами B и A , что означает, что тема A зависит от темы B ($1 \leq A \neq B \leq N$).

Формат выходных данных

Выведите искомый порядок в единственную строку выходного файла: N целых чисел, обозначающих номера тем. Если в темах имеется циклическая зависимость и искомого порядка не существует, выведите слова “bad course” вместо порядка.

Пример

input.txt	output.txt
5 7 1 5 1 3 5 2 5 4 3 4 1 2 4 2	1 3 5 4 2
3 3 1 2 2 3 3 2	bad course