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 возможный, 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 ($1 \leqslant N \leqslant 400$, $1 \leqslant M \leqslant \frac{1}{2}N(N-1)$).

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

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:	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 возможный, 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 ($1 \leqslant N \leqslant 5\,000$, $1 \leqslant M \leqslant 100\,000$).

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

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 3. Наложение рамок

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

Рассмотрим следующие рамки, расположенные в массивах размерностью 9×8.

```
........    ........    ........    ........    .CCC....
EEEEEE..    ........    ........    ..BBBB..    .C.C....
E....E..    DDDDDD..    ........    ..B..B..    .C.C....
E....E..    D....D..    ........    ..B..B..    .CCC....
E....E..    D....D..    ....AAAA    ..B..B..    ........
E....E..    D....D..    ....A..A    ..BBBB..    ........
E....E..    DDDDDD..    ....A..A    ........    ........
E....E..    ........    ....AAAA    ........     ........
EEEEEE..    ........    ........    ........     ........
   1           2           3           4           5
```

Теперь наложим их одна на другую, начиная с первой внизу и заканчивая пятой наверху. Если часть рамки закрывает другую, то она скрывает нижнюю часть. Полученная стопка рамок будет выглядеть следующим образом:

```
.CCC....
ECBCBB..
DCBCDB..
DCCC.B..
D.B.ABAA
D.BBBB.A
DDDDAD.A
E...AAAA
EEEEEE..
```

В каком порядке рамки складывались, начиная снизу? Ответом будет EDABC.

Ваша задача определить порядок, в каком рамки накладывались друг на друга, начиная снизу по заданной картинке результирующей стопки рамок.

Правила следующие: Ширина рамки — это одна литера, а стороны рамки не могут быть короче 3 литер. По крайней мере, часть каждой из четырех сторон рамки можно увидеть. Угол показывает две стороны. Рамки обозначаются заглавными латинскими буквами. Разные рамки обозначаются разными буквами.

Формат входных данных

Каждый блок входного файла содержит высоту h на первой строке и ширину w на второй ($h < 10, w < 10$). Далее дана картинка сложенных рамок в виде h строк по w литер в каждой.

Ваш входной файл может содержать несколько тестов, заданных в описанном выше формате, без пустых строк между тестами. Все тесты из входного файла должны обрабатываться последовательно.

Формат выходных данных

В выходной файл в качестве решения записывается строка, состоящая из букв рамок в той последовательности, в которой они складывались, начиная снизу. Если существует несколько

вариантов их расположения, нужно перечислить все варианты в алфавитном порядке, каждый на отдельной строке. Всегда существует, по крайней мере, один вариант расположения рамок для каждого теста. Пустых строк не вставлять.

Пример

input.txt	output.txt
9 8 .CCC.... ECBCBB.. DCBCDB.. DCCC.B.. D.B.ABAA D.BBBB.A DDDDAD.A E...AAAA EEEEEE..	EDABC