

Задача А. Матрица расстояний

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

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами. Гарантируется, что в графе нет циклов отрицательного веса.

Формат входных данных

В первой строке вводится единственное число N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (j -ое число в i -ой строке соответствует весу ребра из вершины i в вершину j). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

Формат выходных данных

Выведите N строк по N чисел — матрицу кратчайших расстояний между парами вершин. j -ое число в i -ой строке должно быть равно весу кратчайшего пути из вершины i в вершину j .

Пример

<code>matrix.in</code>	<code>matrix.out</code>
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

Задача В. Максимальное расстояние

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

Дан ориентированный взвешенный граф. Вам необходимо найти пару вершин, кратчайшее расстояние от одной из которых до другой максимально среди всех пар вершин.

Формат входных данных

В первой строке вводится единственное число N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы — всегда нули.

Формат выходных данных

Выведите искомое максимальное кратчайшее расстояние.

Пример

<code>maxdist.in</code>	<code>maxdist.out</code>
4	16
0 5 9 -1	
-1 0 2 8	
-1 -1 0 7	
4 -1 -1 0	

Задача С. Существование кратчайшего пути

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

Дан ориентированный взвешенный граф. По его матрице смежности нужно для каждой пары вершин определить, существует ли кратчайший путь между ними или нет.

Комментарий: Кратчайший путь может не существовать по двум причинам:

- Нет ни одного пути
- Есть пути сколь угодно маленького веса

Формат входных данных

В первой строке входного файла записано единственное число: N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках по N чисел — матрица смежности графа (j -е число в i -й

строке соответствует весу ребра из вершины i в вершину j): число 0 обозначает отсутствие ребра, а любое другое число — наличие ребра соответствующего веса. Все числа по модулю не превышают 100.

Формат выходных данных

Выведите N строк по N чисел. j -е число в i -й строке должно соответствовать кратчайшему пути из вершины i в вершину j . Число должно быть равно 0, если пути не существует, 1, если существует кратчайший путь, и 2, если пути существуют, но бывают пути сколь угодно маленького веса.

Пример

<code>exists.in</code>	<code>exists.out</code>
5	1 1 1 0 0
0 1 2 0 0	1 1 1 0 0
1 0 3 0 0	1 1 1 0 0
2 3 0 0 0	0 0 0 2 2
0 0 0 4 -1	0 0 0 2 2
0 0 0 -1 0	

Задача D. Есть ли цикл?

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

Дан ориентированный граф. Требуется определить, есть ли в нем цикл.

Формат входных данных

В первой строке вводится число вершин $N \leq 50$. Далее в N строках следуют по N чисел, каждое из которых — 0 или 1. j -ое число в i -ой строке равно 1 тогда и только тогда, когда существует ребро, идущее из i -ой вершины в j -ую. Гарантируется, что на диагонали матрицы будут стоять нули.

Формат выходных данных

Выведите 0, если в заданном графе цикла нет, и 1, если он есть.

Пример

<code>cycle.in</code>	<code>cycle.out</code>
3	1
0 1 0	
0 0 1	
1 0 0	
3	0
0 1 1	
0 0 0	
0 1 0	