

Задача А. Дейкстра

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

Дан ориентированный взвешенный граф. Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входного файла

В первой строке входного файла три числа: N , S и F ($1 \leq N \leq 2000, 1 \leq S, F \leq N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках по N чисел — 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 выходного файла

Вывести искомое расстояние или -1 , если пути не существует.

Пример

dijkstra.in	dijkstra.out
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача В. Расстояние между вершинами

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

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входного файла

Первая строка входного файла содержит натуральные числа N , M , S и F ($N \leq 70\,000$, $M \leq 200\,000$, $1 \leq S, F \leq N$, $S \neq F$) — количество вершин и рёбер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие M строк содержат по три натуральных числа b_i , e_i и w_i — номера концов i -ого ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Формат выходного файла

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами S и F . Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода.

Если пути из S в F не существует, выведите -1 .

Пример

distance.in	distance.out
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3 1 2 3

Задача С. Кратчайший путь

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

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u .

Формат входного файла

Первая строка входного файла содержит n , m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leq n \leq 2000$, $1 \leq m \leq 5000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходного файла

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u , '*' если не существует пути из s в u и '-' если не существует кратчайший путь из s в u .

Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	