

# Задача из Депутатского

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 1 секунда         |
| Ограничение по памяти:  | 256 мегабайт      |

Глава посёлка Депутатский, разглядывая в очередной раз карту области, восхищался коллегами, управляющими Берлином, Парижем, Варной и Лейпцигом. И тут ему в голову пришла гениальная идея: переименовать посёлок в Манхэттен. Однако чтобы общественные обсуждения прошли гладко, он решил начать с воссоздания известной дорожной сети Манхэттена.

Итак, он поручил снести все здания в поселке и начать укладывать дорогу. Через некоторое время работа была сделана, но, начав сверять результат с генеральным планом, он пришёл в ужас!

Дорожная система сейчас представляет собой сетку размера  $n \times m$ : в каждом из  $n$  рядов находится по  $m$  перекрестков. Обозначим перекресток на пересечении  $r$ -й строки и  $c$ -го столбца в индексации с нуля за  $(r, c)$ . Тогда сейчас построены двусторонние дороги, соединяющие перекрёстки  $(i_1, j_1)$  и  $(i_2, j_2)$ , если  $|i_1 - i_2| + |j_1 - j_2| = 1$ . Также были построены двусторонние дороги, соединяющие первый и последний перекресток в каждой строке и столбце, то есть соединёнными оказались следующие пары перекрёстков:  $(i, 0)$  и  $(i, m - 1)$  для всех  $0 \leq i < n$ , а также  $(0, j)$  и  $(n - 1, j)$  для всех  $0 \leq j < m$ . Наконец, укладку дороги на некоторых перекрёстках ещё не закончили, поэтому сейчас невозможно проехать по дороге, соединяющей такой перекрёсток с любым другим.

После увиденного глава посёлка хочет задать взбучку главе стройки, поэтому сказал ехать ему навстречу. Пусть первый сейчас находится на перекрестке  $(i_1, j_1)$ , а второй — на перекрёстке  $(i_2, j_2)$ . За одну минуту они могут **одновременно** переместиться одним из следующих образов:

- Первый окажется в  $((i_1 + 1) \bmod n, j_1)$ , второй в  $((i_2 - 1 + n) \bmod n, j_2)$ .
- Первый окажется в  $((i_1 - 1 + n) \bmod n, j_1)$ , второй в  $((i_2 + 1) \bmod n, j_2)$ .
- Первый окажется в  $(i_1, (j_1 + 1) \bmod m)$ , второй в  $(i_2, (j_2 - 1 + m) \bmod m)$ .
- Первый окажется в  $(i_1, (j_1 - 1 + m) \bmod m)$ , второй в  $(i_2, (j_2 + 1) \bmod m)$ .

При этом, если хотя бы один из перекрёстков, куда они хотят переместиться, ещё не достроен, то такое перемещение невозможно.

Глава посёлка хочет понять, за какое минимальное время он сможет встретиться на одном перекрёстке с главой стройки, если будет действовать оптимально, и смогут ли они встретиться вообще.

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное целое число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных. Далее следует их описание.

В первой строке каждого набора входных данных содержится два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — размеры дорожной системы посёлка.

В следующих  $n$  строках каждого набора входных данных содержится строка длины  $m$ , состоящая из символов «.» и «#». Если перекрёсток  $(i, j)$  достроен, то в  $i$ -й строке в  $j$ -й позиции (в индексации с нуля) находится символ «.». Если перекрёсток не достроен, то соответствующий символ равен «#».

В следующей строке каждого набора входных данных содержится два целых числа  $(a, b)$  ( $0 \leq a \leq n - 1, 0 \leq b \leq m - 1$ ) — перекресток, на котором находится глава посёлка в начале пути.

В последней строке каждого набора входных данных содержится два целых числа  $(c, d)$ , ( $0 \leq c \leq n - 1, 0 \leq d \leq m - 1$ ) — перекрёсток, на котором находится глава стройки в начале пути.

Гарантируется, что перекрестки  $(a, b)$  и  $(c, d)$  достроены, а также, что эти перекрестки различны.

Гарантируется, что сумма  $n \cdot m$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

## Формат выходных данных

Выведите одно целое число — минимальное количество минут, по прошествии которых глава посёлка и стройки смогут встретиться. Если встреча невозможна, выведите  $-1$ .

## Пример

| стандартный ввод                                                                               | стандартный вывод |
|------------------------------------------------------------------------------------------------|-------------------|
| 2<br>5 5<br>..#..<br>..#..<br>#####<br>..#..<br>..#..<br>0 0<br>3 3<br>1 2<br>..<br>0 0<br>0 1 | 2<br>-1           |

## Замечание

В первом наборе входных данных первое одновременное перемещение может быть следующим: глава посёлка едет налево, проезжает по дороге, соединяющей перекрёстки  $(0, 0)$  и  $(0, 4)$ , и оказывается на перекрёстке  $(0, 4)$ ; глава стройки, соответственно, едет направо и оказывается на перекрёстке  $(3, 4)$ . В качестве второго одновременного перемещения глава посёлка едет вверх и проезжает по дороге, соединяющей перекрёстки  $(0, 4)$  и  $(4, 4)$ , оказываясь на перекрёстке  $(4, 4)$ ; глава стройки в это время едет вниз и тоже оказывается на перекрёстке  $(4, 4)$ . Таким образом они встретятся за 2 перемещения.

Во втором наборе входных данных можно показать, что глава посёлка и глава стройки никогда не встретятся.