

## Problem J. Just an Edit Distance

Input file: *standard input*  
 Output file: *standard output*  
 Time limit: 5 seconds  
 Memory limit: 1024 mebibytes

You have a table with  $H$  rows and  $W$  columns. Each cell of the table contains a letter.

You are going to construct a string by the following steps.

1. Pick an arbitrary starting cell in the table as the current cell.
2. Let  $S$  be a string of length 1 containing the letter in the starting cell.
3. Do either:
  - stop building  $S$ , or
  - select a cell which shares an edge with the current one (there are up to four such cells). Then move to this cell (make it the current one) and append the letter in this cell to  $S$ . After that, repeat step 3.

You also have a string  $T$ . Your mission is to minimize the edit distance between  $S$  and  $T$ .

*Edit distance* (also known as Levenshtein distance) between strings  $U$  and  $V$  is the minimum number of steps required to convert  $U$  into  $V$ . Each step is one of the following:

- Replace a character in  $U$  with another one.
- Insert a character into  $U$ .
- Delete a character from  $U$ .

### Input

The first line of input contains two integers,  $H$  and  $W$ : the height and the width of the table ( $2 \leq H, W \leq 100$ ).

Each of the next  $H$  lines contains  $W$  lowercase English letters. Together, they form the table.

The next line contains the string  $T$  consisting of lowercase English letters ( $1 \leq |T| \leq 2000$ ).

### Output

Output a line with a single integer: the minimum possible edit distance between  $S$  and  $T$ .

### Example

<i>standard input</i>	<i>standard output</i>
2 2 ab ar abracadabra	2