String Merging | Problem Code: **STRMRG**

For a string **S**, let's define a function **F**(**S**) as the minimum number of blocks consisting of consecutive identical characters in **S**. In other words, **F**(**S**) is equal to 1 plus the number of valid indices **i** such that $S_i \neq S_{i+1}$.

You are given two strings **A** and **B** with lengths **N** and **M** respectively. You should merge these two strings into one string **C** with length **N+M**. Specifically, each character of **C** should come either from **A** or **B**; all characters from **A** should be in the same relative order in **C** as in **A** and all characters from **B** should be in the same relative order in **C** as in **B**.

Compute the minimum possible value of **F**(**C**).

Input

- The first line of the input contains a single integer **T** denoting the number of test cases. The description of **T** test cases follows.
- The first line of each test case contains two space-separated integers **N** and **M**.
- The second line contains a single string **A** with length **N**.
- The third line contains a single string **B** with length **M**.

Output

For each test case, print a single line containing one integer — the minimum possible value of **F**(**C**).

Constraints

- $1 \le T \le 100$
- $1 \le N, M \le 5{,}000$
- $1 \le$ sum of **N** in all test cases $\le 5{,}000$
- $1 \le$ sum of **M** in all test cases $\le 5{,}000$
- strings **A**, **B** consist only of lowercase English letters

Subtasks

Subtask 1 (10 points): $1 \le T, N, M \le 10$

Subtask 2 (20 points): the characters of **A** and **B** are sorted in non-decreasing order

Subtask 3 (70 points): original constraints

Example

```
Input:


1
4 4
abab
baba

Output:


5
```

Explanation

Example case 1: One possible way to choose the string **C** to get the desired answer is "abbaabba".

Author:	6★ kingofnumbers (/users/kingofnumbers)
Date Added:	2-01-2018
Time Limit:	1 secs
Source Limit:	50000 Bytes
Languages:	ADA, ASM, BASH, BF, C, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 6.3, CPP14, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAVA, JS, kotlin, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYPY, PYTH, PYTH 3.5, RUBY, rust, SCALA, SCM chicken, SCM guile, SCM qobi, ST, swift, TCL, TEXT, WSPC