Hash Code

Pizza

Practice Problem for Hash Code

Introduction

Did you know that at any given time, someone is cutting pizza somewhere around the world? The decision about how to cut the pizza sometimes is easy, but sometimes it's *really* hard: you want just the right amount of tomatoes and mushrooms on each slice. If only there was a way to solve this problem using technology...

Problem description

Pizza

The pizza is represented as a rectangular, 2-dimensional grid of ***R*** rows and ***C*** columns. The cells within the grid are referenced using a pair of 0-based coordinates $[r, c]$, denoting respectively the row and the column of the cell.

Each cell of the pizza contains either:

- mushroom, represented in the input file as `M`; or
- tomato, represented in the input file as `T`

Slice

A slice of pizza is a rectangular section of the pizza delimited by two rows and two columns, without holes. The slices we want to cut out must contain at least ***L*** cells of each ingredient (that is, at least ***L*** cells of mushroom and at least ***L*** cells of tomato) and at most ***H*** cells of any kind in total - surprising as it is, there is such a thing as too much pizza in one slice.

The slices being cut out cannot overlap. The slices being cut do not need to cover the entire pizza.

Goal

The goal is to cut correct slices out of the pizza maximizing the total number of cells in all slices.

Input data set

The input data is provided as a data set file - a plain text file containing exclusively ASCII characters with lines terminated with a single '\n' character at the end of each line (UNIX-style line endings).

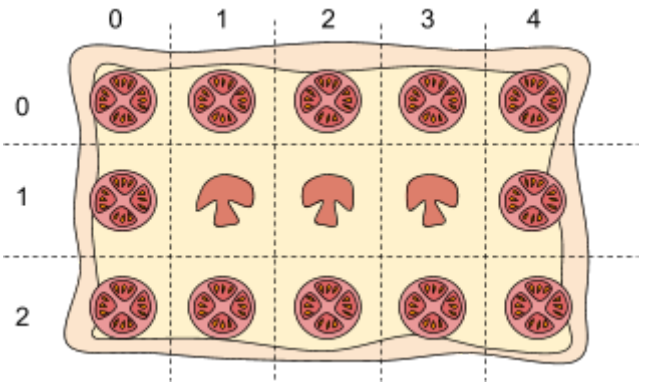
File format

The file consists of:

- one line containing the following natural numbers separated by single spaces:
  - ***R*** $(1 \leq R \leq 1000)$ is the number of rows,
  - ***C*** $(1 \leq C \leq 1000)$ is the number of columns,
  - ***L*** $(1 \leq L \leq 1000)$ is the minimum number of each ingredient cells in a slice,
  - ***H*** $(1 \leq H \leq 1000)$ is the maximum total number of cells of a slice

- ***R*** lines describing the rows of the pizza (one row after another). Each of these lines contains ***C*** characters describing the ingredients in the cells of the row (one cell after another). Each character is either 'M' (for mushroom) or 'T' (for tomato).

Example



3 5 1 6 TTTTT TMMMT TTTTT	3 rows, 5 columns, min 1 of each ingredient per slice, max 6 cells per slice

Example input file.

Submissions

File format

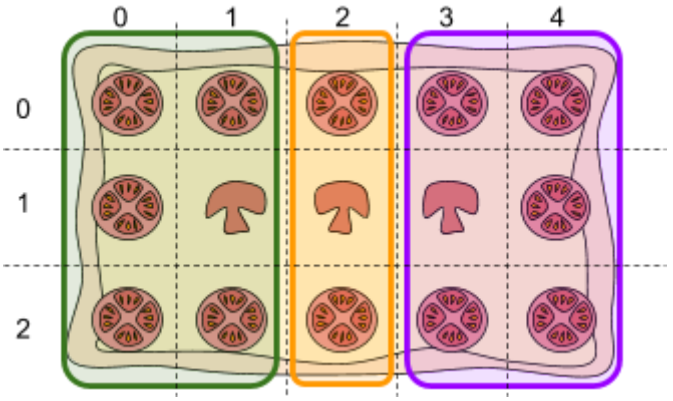
The file must consist of:

- one line containing a single natural number ***S*** $(0 \leq S \leq R \times C)$, representing the total number of slices to be cut,
- ***S*** lines describing the slices. Each of these lines must contain the following natural numbers separated by single spaces:
  - ***r*****1**, ***c*****1**, ***r*****2**, ***c*****2** $(0 \leq r_1, r_2 < R, 0 \leq c_1, c_2 < C)$ describe a slice of pizza delimited by the rows ***r*****1** and ***r*****2** and the columns ***c*****1** and ***c*****2**, including the cells of the delimiting rows and columns. The rows (***r*****1** and ***r*****2**) can be given in any order. The columns (***c*****1** and ***c*****2**) can be given in any order too.

Example

3 0 0 2 1 0 2 2 2 0 3 2 4	3 slices. First slice between rows (0,2) and columns (0,1). Second slice between rows (0,2) and columns (2,2). Third slice between rows (0,2) and columns (3,4).

Example submission file.



Slices described in the example submission file marked in green, orange and purple.

Validation

For the solution to be accepted:

- the format of the file must match the description above,
- each cell of the pizza must be included in at most one slice,
- each slice must contain at least ***L*** cells of mushroom,
- each slice must contain at least ***L*** cells of tomato,
- total area of each slice must be at most ***H***

Scoring

The submission gets a score equal to the total number of cells in all slices.

Note that there are multiple data sets representing separate instances of the problem. The final score for your team is the sum of your best scores on the individual data sets.

Scoring example

The example submission file given above cuts the slices of 6, 3 and 6 cells, earning 6 + 3 + 6 = 15 points.