

Problem Statement for Trial Round (2014-04-04)

Painting the Facade



1. Introduction

The day has come to paint a huge mural on the facade of Google Paris office.

A picture has been decided on and a specialized machine has been hired to perform the painting. Unfortunately, it turns out that the painting operations supported by the machine are quite low-level.

Therefore before putting the machine to work, the target picture has to be translated into a list of instructions supported by the machine.

2. Task

Given the target picture, come up with a list of commands that produce it using as few commands as possible.

3. Problem description

3.1. Picture

The picture is a rectangular grid of square cells, each of which either has to be painted in black, or has to remain clear. At the beginning, the entire wall (all cells) is clear.

The cells of the picture are referred to using their coordinates, where (R, C) denotes a cell in R -th row and C -th column of the picture. Indexing of the rows and columns is 0-based, with the cell $(0, 0)$ located in the top-left corner of the picture.

3.2. Painting

The machine supports the following commands:

- PAINTSQ R C S - paints all cells within the square of $(2S+1) \times (2S+1)$ dimensions centered at (R, C) . In particular, command “PAINTSQ R C 0” paints a single cell (R, C) . For the command to be valid, the entire square has to fit within the dimensions of the painting.
- ERASECELL R C - unpaints the cell (R, C)

4. Input data

The input data is provided as a plain text file containing exclusively ASCII characters with lines terminated with UNIX-style line endings (single ‘\n’ character ending each line).

The file consists of:

- one line containing the following natural numbers separated by single spaces:
 - **N** denotes the number of rows of the picture
 - **M** denotes the number of columns of the picture
- **N** subsequent lines describing individual rows of the picture. The i -th ($0 \leq i < N$) such line contains **M** characters describing picture cells in consecutive columns of the i -th row of the picture, starting with the column 0. Each character is either:
 - ‘.’ - denoting a cell that has to remain clear
 - ‘#’ - denoting a cell that has to be painted

Example of input file

```
5 7
....#..
..###..
..#.#..
..###..
..#....
```

5. Submissions

5.1. File format

Team submission needs to be described in a plain-text ASCII file with either Unix-style or Windows-style line endings.

The file needs to start with one line containing a single natural number **S** representing the number of instructions provided for the machine, with $(0 \leq S \leq NM)$. Then individual instructions should follow - each in separate line.

Example of submission file

```
4 // Four instructions.
PAINTSQ 3 3 1
PAINTSQ 0 4 0 // Paint a single cell.
PAINTSQ 5 2 0 // Paint a single cell.
ERASECELL 3 3 // Unpaint the middle cell.
```

5.2. Validation

For the solution to be accepted, it has to meet the following criteria:

- the format of the file has to match the description above

- the provided list of instructions has to produce the picture specified in the input file. If the resulting picture differs from the target one, the solution will not be accepted.

5.3. Scoring

Each valid submission will be immediately scored and the score will be revealed to the team. The score of a valid submission is the number of instructions provided. The goal is to minimize this score.

The teams are allowed to submit multiple solutions - the best valid solution from each team will be used for team ranking.

Teams will be ranked according to their best submission score. In an event of a tie (two teams having the same best submission score), the team that reached that score for the first time earlier will be ranked higher. Resubmitting the same best solution again does not hurt the teams ranking.

© Google, 2014. All rights reserved.

Pictures: Eric Laignel.