¿Cuán uniforme es la imagen?

Contribución de Sergio Porter

Descripción del problema

Se tiene una imagen en blanco y negro, cada pixel de la misma queda representada por un bit. La ubicación de un pixel se puede describir con un par de coordenadas, ambas en el rango [1,N]. Dos pixeles son adyacentes si tienen un lado en común.

Una herramienta como "paint" divide una imagen en regiones uniendo pixeles del mismo color que sean adyacentes. Dos pixeles están en una misma región si tienen un mismo color y existe una cadena de pixeles adyacentes, también del mismo color, que los una. Finalmente una región es adyacente a otra si se puede designar un par de pixeles, uno por región que sean adyacentes.

Hay comandos que permiten operar sobre regiones cambiándoles el color, lo que vale también, aunque poco usado, para el blanco y el negro. Toda región negra adyacente a una blanca que pasa a negra se fusiona con la misma. Y recíprocamente. Como consecuencia de ello baja el número de regiones y la región repintada adquiere una superficie suma de las superficies de las regiones que se han fusionado.

Repitiendo suficiente número de veces el repintado de alguna región se puede conseguir que todos los pixeles sean iguales, sin importar si son ceros o unos.

Para tener una medida de cuan uniforme o variado es un mapa de bits se ha decidido cargarlo en "paint" y medir la cantidad de veces que hay que repintar una región para lograr que el cuadrado sea todo blanco o todo negro.

Para que la medida sea repetible se ha decidido que la región que invierte de blanco a negro o viceversa sea la que contiene el bit de la fila 1 y columna 1. Esa región en cada maniobra se irá agrandando.

Se quiere conocer por medio de un programa **uniformar.c**, **uniformar.cpp** o **uniformar.pas** la cantidad de acciones de repintado que permita alcanzar un cuadro uniforme de bits.

Datos de entrada

Se recibe un archivo **uniformar.in** con el siguiente formato:

- Una línea con un entero: el valor de **N** (**1** ≤ **N** ≤ **2.000**) el tamaño de la cuadrícula.
- **N** líneas, cada una con **N** caracteres. Estos caracteres sólo pueden ser dígitos **0** y **1**. El primer carácter de cada línea corresponde a la columna 1 y así siguiendo.

Datos de salida

Se debe generar un archivo **uniformar.out** conteniendo:

- Una línea con un entero P, la mínima cantidad de acciones de cambio que uniforman el cuadro.

Ejemplo

Si la entrada **uniformar.in** fuera:

```
6
001100
010101
110110
110110
011001
110011
```

La salida **uniformar.out** debería ser:

```
3
```

El siguiente cuadro ilustra el progreso alcanzado en cada cambio:

```
111100  000000  111111
110101  000001  111111
110110  000000  111111
110110  000000  111111
011001  000001  111111
110011  000011  111111
```