ColorGrid (Recursion) FRQ

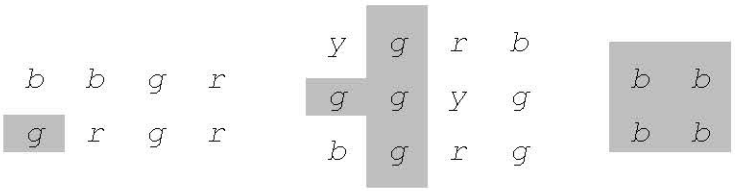
A *color grid* is defined as a two-dimensional array whose elements are character strings having values "b" (blue), "r" (red), "g" (green), or "y" (yellow). The elements are called pixels because they represent pixel locations on a computer screen. For example,

```
b  b  g  r
g  r  g  r

r  r  r  r  r

y  g  r
b  y  g
g  r  b
b  b  g
```

A connected region for any pixel is the set of all pixels of the same color that can be reached through a direct path along horizontal or vertical moves starting at that pixel. A connected region can consist of just a single pixel or the entire color grid. For example, if the two dimensional array is called `pixels`, the connected region for `pixels[1][0]` is as shown here for three different arrays.



The class `ColorGrid`, whose declaration is shown below, is used for storing, displaying, and changing the colors in a color grid.

```
public class ColorGrid
{
  private String[][] pixels;
  private int rows;
  private int cols;

  /** Creates numRows x numCols ColorGrid from String s. */
  public ColorGrid(String s, int numRows, int numCols)
  { /* to be implemented in part (a) */ }

  /* If 0 <= row < rows and 0 <= col < cols, paints the
   * connected region of pixels[row][col] the newColor.
   * Does nothing if oldColor is the same as newColor.
   * Precondition:
   * - pixels [row] [col] is oldColor, one of "r", "b", "g", or "y".
   * - newColor is one of "r", "b", "g", or "y".
   */
  public void paintRegion(int row, int col, String newColor,
  String oldColor)
  { /* to be implemented in part (b) */ }

  //Other methods are not shown.
}
```

ColorGrid (Recursion) FRQ

(a) Write the implementation code for the `ColorGrid` constructor. The constructor should initialize the pixels matrix of the *ColorGrid* as follows: The dimensions of pixels are `numRows * numCols`. `String s` contains `numRows * numCols` characters, where each character is one of the colors of the grid - `"r"`, `"g"`, `"b"`, or `"y"`. The characters are contained in `s` row by row from top to bottom and left to right. For example, given that `numRows` is `3`, and `numCols` is `4`, if s is `"brrygrggyyyr"`, pixels should be initialized to be

b	r	r	y
g	r	g	g
y	y	y	r

Complete the following constructor:

```
/** Creates numRows x numCols ColorGrid from String s. */
public ColorGrid(String s, int numRows, int numCols)
```

Testing Code

(b) Write the implementation of the `paintRegion` method as started below. Note: You must write a recursive solution. The `paintRegion` paints the connected region of the given pixel, specified by `row` and `col`, a different color specified by the `newColor` parameter. If `newColor` is the same as `oldColor`, the color of the given pixel, `paintRegion` does nothing. To visualize what `paintRegion` does, imagine that the different colors surrounding the connected region of a given pixel form a boundary. When paint is poured onto the given pixel, the new color will fill the connected region up to the boundary.

For example, the effect of the method call `c.paintRegion(2, 3, "b", "r")` on the `ColorGrid c` is shown here. (The starting pixel is shown in a frame, and its connected region is shaded.)

before

r	r	b	g	y	y
b	r	b	y	r	r
g	g	r	r	r	b
y	r	r	y	r	b

after

r	r	b	g	y	y
b	r	b	y	b	b
g	g	b	b	b	b
y	b	b	y	b	b

Complete the method `paintRegion` below. Note: Only a recursive solution will be accepted.

```
/* If 0 <= row < rows and 0 <= col < cols, paints the
 * connected region of pixels[row][col] the newColor.
 * Does nothing if oldColor is the same as newColor.
 * Precondition:
 * - pixels[row][col] is oldColor, one of "r", "b", "g", or "y".
 * - newColor is one of "r", "b", "g", or "y".
 */
public void paintRegion(int row, int col, String newColor,
                        String oldColor)
```

Testing Code