Lecture 23: Arrays and Images, Part 2

Nov 1, 2019

Announcements

- Should have received Labs 5 and 6 now
- HW 6 due Monday
- TA/Tutor Applications due soon. Consider applying!
- Lab 8: Try to write your own implementation plan!

Last Time

- Learned about arrays
- Learned about for loops

Recap: Arrays for For Loops

- for (int x = 0; x < array.length; x = x+1) { ... }
  - First parameter: assign start value of loop variable
  - Second parameter: loop condition (just like while)
  - Third parameter: how the loop variable changes
- Can easily convert while loops to for loops
- Recall that array.length tells you the size of the array
- Note: array.length is not a method call! (It is not array.length())
Today’s Plan

- 1-D Array Example: LetterHistogram
- 2-D Arrays
- Lab 7: SImages, Files, ...
- Image Manipulation

Examples

- Sorting and swapping: MoreStatsWithFor
- Making histograms: TextAnalyzer

Moving on:
2-Dimensional Arrays

What kind of object is int[][] pixels?

- int[] grades; // an array of ints
- String[] months; // an array of Strings
- JButton[] choices; // an array of JButtons
- AnyJavaClass[] xyz; // an array of AnyJavaClass
- int[][] pixels; // an array of int[]s!

Digital Images

A discrete collection of brightness intensity values (ints)
Representing Intensities

- Black & white image: collection of gray-scale values (e.g., 0-255) in a rectangular array
- int [][] pixels = myImage.getPixelArray();
- pixel[x][y] is value in x\textsuperscript{th} column (from left), y\textsuperscript{th} row from top
- pixel[0][0] is top left value

Visualizing Arrays

Note: It's not matrix notation!

Array of arrays

- Not all of the 1-D arrays need same length in 2-D array
- String[][] wordTable can have wordTable[i].length ≠ wordTable[j].length
- wordTable.length is number of 1-D arrays in the 2-D array!
- Can also have more than 2 dimensions:
  - int [][][] colorPixels;
  - colorPixels[0] : 2-D array of RED values
SImages, Files, and Layouts

- SImage Class
- JFileChooser
- BorderLayout
- Sample Code: SimpleImage

Image Transformations

- Quantize
- Brightness
Edges

- Look at Edges in action
- We need to find edges in pixel array, but how?
  - Look at neighboring pixels
  - If difference in brightness is significant, mark current pixel as black (it's an edge!)
  - Otherwise, mark it white

Shift

What other things can we do with arrays?

Copy them to “shift” the values
Another example: Scaling
8 \times 70 / 100 = 5.6 = 5
8 \times 10 = 80
11 \times 14
8 \times 140 / 100 = 11.2 = 11

scaled[0][0] = orig[0*100/70][0*100/70] = orig[0][0]
scaled[0][1] = orig[0*100/70][1*100/70] = orig[0][1]
scaled[0][2] = orig[0*100/70][2*100/70] = orig[0][2]