CS 114, Fall 2017, Prof. Calvin
Problem set #10
Due: In class, Tuesday Nov. 14.

The purpose of this assignment is to explore an application of selection. The goal is to "un-clutter" a set of images. The Birds directory on moodle contains a set of seventeen files, birds1.ppm through birds17.ppm, storing images in the PPM (Portable Pixel Map) format. The background is similar for each image, but each has some clutter (attacking birds). The directory also contains Blur.java, which has the boilerplate code to open and read in the files, and also write an output file (it just averages the pixel values over the images). You need to write a program (call it Clear.java) that produces an image file with the clutter removed.

The images are sampled from a scene in the Alfred Hitchcock move "The Birds"; you can see the clip at `https://www.youtube.com/watch?v=hplpQt424Ls`. Promise that you won't use the image files for commercial entertainment purposes.

The PPM files are plain text files with the following format. The first line gives the image type, which is always "P3" for this exercise. The next line has two integers giving the number of columns and rows of the image. The next line has an integer giving the maximum level for any color. Then the rest of the file is a sequence of integers, three for each pixel, giving the red, green, and blue intensities.

For example, consider the following image file:

```
P3
3 4
255
255 0 0 0 255 0 0 0 255
0 0 0 255 255 255 120 120 120
255 255 0 255 0 255 0 255 255
100 100 100 200 200 200 0 100 0
```

The first three lines are the header, with the first line indicating that this is a P3 (plain) image. The next line tells us that this image has 3 columns and 4 rows of pixels. The next line gives the maximum intensity level of 255; all subsequent numbers in the file are between and 0 and the maximum 255. The following lines give the red, green, blue values for each pixel; the first pixel has intensities 255 (red), 0 (green) and 0 (blue), so the top left pixel is red. The next pixel is green and the last pixel on the first row is blue. The first pixel on the second row is black (all three intensities equal to zero).

On the Linux machines in the OSL lab, you can display a ppm image by clicking on it. On a Windows machine, you can display PPM images using, for example, the LibreOffice Draw program or the ppmReader.html file in the Birds directory (open it with the Chrome browser).

Be prepared to demonstrate your program during our next class. Your program should produce an image that shows what the view from the car would be if there were no birds.

Exercises:

(1) Consider a version of the problem described above with n images, each having P pixels, with pixels having values between 1 and C. Briefly describe an efficient algorithm to solve this problem when n is large compared to P and C, and characterize the running time of your algorithm in terms of n, P, and C. State whether your analysis is average or worst case.

(2) Suppose you need to sort a list of n integers that is "almost" sorted to begin with in the sense that if the ith smallest element is at index j, then $|i - j| \leq k$, for some fixed number k (small compared to n). Describe an efficient algorithm and characterize the asymptotic running time (state whether your analysis is worst-case or average-case).

(3) Describe how to sort n numbers, each between 0 and $n^2 - 1$, in time $O(n)$. (Hint: how could you apply radix sort?)

(4) Suppose that we perform a mixture of n operations on an initially empty priority queue. Argue that in the worst case, these n operations take time $\Omega(n \lg(n))$. Assume that the order of elements in the priority queue is determined by pairwise comparisons, but make no other assumptions about how the priority queue is implemented.

(5) We used a binary decision tree to model a sorting algorithm; each comparison had two possible outcomes. The Java compareTo method gives three outcomes: less, equal, or greater. A sorting algorithm using three outcomes would be modeled with a ternary tree; each internal node has three children. Derive a lower bound for a sorting algorithm that uses such three-way comparisons.

(6) You are given n red and n blue water jugs, all of different shapes and sizes. All the red jugs hold different amounts of water, as do the blue jugs. For every red jug, there is exactly one blue jug that holds the same amount of water, and vice versa.

Your task is to group the jugs into pairs of red and blue jugs that hold the same amount of water. To do so, you may perform the following operation: pick a red jug and a blue jug, fill the red jug with water, then pour the water from the red jug into the blue jug. This operation will tell you either that the red jug holds less, the same, or more water than the blue jug. You may not compare two red or two blue jugs. You want to determine the grouping using the fewest comparisons possible.

a) Describe a deterministic algorithm that uses $\Theta(n^2)$ comparisons.

b) Argue that any algorithm solving this problem must make $\Omega(n \lg(n))$ comparisons in the worst case.

c) Give a randomized algorithm making on average $O(n \lg(n))$ comparisons.