Chapter 15

Graphics file Formats

Exercises for this chapter deal with the features of raster image files, methods for color reduction, and techniques for file compression.

Exercises

15-1. Input for this program is an image file and the integer reduction factor. The set of color values in the image file is then replaced with a reduced set of colors, as illustrated in fig. 15-1. Display a given image file using various reduction factors and observe the relative effects for the three RGB components.

15-2. This program is essentially the same as the program for the preceding exercise. Dividing the number of color values in an image file by the desired number of reduction levels produces the reduction factor for the image file.

15-3. For this exercise, various sets of reduction numbers for the RGB color components can be input and the relative picture quality evaluated.

15-4. Given an input integer value k, the k most frequently occurring colors are used in the display. The input image file can first be processed to reduce the bit representation for the individual RGB components, and picture quality can be evaluated for different values of parameter k.

15-5. As illustrated in Fig. 15-2, this program subdivides the color space into k subregions, using the average color for each of the subregions to display the input image file. Picture quality for a given input file can be compared for various values of the input parameter k.

15-6. The list of input integers can be provided in a data file, and the program for this exercise scans the input list, searching for adjacent repetitions, as illustrated in Section 15-3. An output file is produced that lists each unique input-file value and the repetition factor for each value.

15-7. As an option, both the number of scan lines and the number values per scan line could be specified as input parameters for the run-length encoding program.

15-8. This program is a modification of the Exercise 15-6 program. A search is to be made for repeated three-element patterns in an input set containing 1,024 integer values for each of the three *RGB* color components. For each color component, the program could first search for a repeated value, then for repeated two-element patterns, then for repeated three-element patterns. A simplified code value is then assigned to each repeated pattern and the remaining integer sequences in the input file. Output from this program is to include a code dictionary (listing the set of color values corresponding to each code) and an output file that lists the code for each repeated three-element pattern, along with the assigned code for each remaining integer sequence in the input file, as illustrated in Section 15-3.

15-9. As an option, both the number of scan lines and the number values per scan line could be specified as input parameters for the LZ encoding program.

15-10. For each *RGB* component in the input file, this program simply counts the number of value occurrences and outputs a table of the frequency counts for each input value, along with the total number of values in the file, as illustrated in Table 15-1.

15-11. Using array or linked-list structures construct a binary tree representation from the output of the program in the previous exercise, as in the example of Figs. 15-3 through 15-8, and generate the compressed image file output.

15-12. The frequency counts from Exercise 15-10 are now used to compute the fraction of the file that is occupied by each color value, and a compressed output file is generated as illustrated by the example in Tables 15-4 and 15-5.

15-13. The calculations for this program are to be performed using Eq. 15-1 with $n = 8$, for each of the three *RGB* components in each of the four groups of pixels in the input list. A transformed output list is to be generated similar to that given in the example in Section 15-3.

15-14. The output from the previous exercise is to be processed through Eq. 15-2 using $n = 8$, and the calculated inverse values can be compared to the input values.

15-15. For this program, a list of output values, such as Table 15-6, could be generated for each integer value of n from 1 to 8.

15-16. This program is a modification of the Exercise 15-13 program, with the input file now processed through Eq. 15-3 using $n = 8$.

15-17. This program is a modification of the Exercise 15-14 program, with the input file now processed through Eq. 15-4 using $n = 8$.

15-18. The program is a modification of the program in the previous exercise. A table of output values could be generated from Eq. 15-4 using integer values for parameter n from 1 to 8.