Data Structures
Assignment 3: Due on Dec 1, 1999 at 9:30am

Figure 1: Some example images of 2D shapes

The goal of this assignment is to familiarize you with the Hash Table data structure. You would be using a hash table to store and retrieve black and white images of 2D shapes such as those shown above in Figure 1. Your program will read in a set of PGM formatted images, which are listed (in no particular order) in an user specified file. You should “insert” these images into the HashTable as they are read. The program should then interact with the user to read in query images and determine if they exist in the HashTable database. The program should answer just yes or no to an query image, along with the number of comparisons done and the computed hash table index.

To implement the HashTable, you are to use

1. Quadratic collision resolution strategy,

2. HashTable size of 2467, and

3. An hash function to map an image to an index as follows.

(a) First, compute the horizontal and vertical projections of the image. The horizontal projection of an image is basically an array whose each cell corresponds to an image row and stores the number of black pixels in that row. Similarly the vertical projection is an array which stores the number of black pixels in each column. An example vertical and horizontal projection is shown in Fig. 2

(b) Second, sort these horizontal and vertical projections in descending order.

(c) Third, concatenate the sorted horizontal and vertical projection arrays. Thus, for the image in Figure 2, concatenation of the sorted horizontal and vertical arrays gives us the number $I_1 = \{6 5 4 4 3 3 3 3 2 2 2 2 2 0 7 7 5 4 4 3 2 2 2 2 2 1 0 0 \}$. Note that this is a 32 digit number. The base of the number for the horizontal projection is the number of columns in the image and the base for the vertical projection digits is the number of rows. In this example, the base for all the digits is 17.

(d) Fourth, perform digit folding on the number $I_1$ – consider groups of five digits at a time, convert them to a base 10 numbers, and then add them up. Let this number be $I_2$. 

1
Figure 2: Horizontal and vertical projection of the example image.

(e) Fifth, select the middle 5 digits from the result of $I_2 \times I_2$. Let this 5 digit number be denoted by $I_3$.
(f) Sixth, take a modulus of $I_3$ with the HashTable Size to arrive at the index.

The coding for this assignment can be divided into the following parts:

1. Modify the Image package (image.c and image.h) to include (i) a function that computes if two images are similar by computing their difference and checking if it is zero, (ii) a function that computes the hash table index using the vertical and horizontal projection given an image and the hash table size.

2. Write a HashTable package (hash.c and hash.h files) that will implement quadratic collision resolution scheme. The package will include the standard procedures that go with a hashed data structure such as functions to add, delete, search, and display the hash table. **The code in hash.c and hash.h might use the functions and data structure in image.c and image.h but should not access the Image data structure directly.**

3. Design a menu based driver that uses the above two packages to implement the 2D shape recognition module. The program should ask the user to read in a set of “database” images into the hash table. Given a new image file name, the program should search for it in the hash table and display its key location and the number of collisions encountered in the process.

**What to submit?** For details about what and how to submit, see Submission Guidelines.