CSCI 3323
Fall 2022
Exam 2

Last Name: _____

First Name: _____

Write answers on the provided answer sheets.

1. (2 pts each) Consider the following variables and their meanings:

   - n = the number of elements in a container
   - m = the number of slots in a hash table
   - α = the load factor of a hash table

   Use the variables above to express the worst-case complexity of each of the following operations:

   (a) left-rotate in a red-black tree

   (b) red-black tree delete (including fixup)

   (c) hash insert (chaining; average case)

   (d) hash minimum (chaining)

   (e) heapify

2. (3 pts each) Briefly define each of the following terms:

   (a) primary clustering

   (b) load factor

   (c) quadratic probing

3. (4 pts) Why would someone consider using a red-black tree instead of using a hash-table? And vice-versa?

4. (4 pts) Compare and contrast chaining and open addressing.

5. Consider the red-black tree given in figure 1.

   (a) (3 pts) Show the order in which nodes would be visited in an inorder traversal.

   (b) (5 pts) Show the steps required to insert 140. Be sure to redraw the tree after applying each case and be sure to specify which case you apply at each step.

   (c) (5 pts) Show the steps required to remove 150. Be sure to specify which cases you apply at each step. Use the original tree as the starting point.

6. Suppose the following array represents a heap: $< 54, 32, 50, 30, 28, 20, 25, 26, 21, 19 >$

   (a) (3 pts) Draw the tree representation of this heap.

   (b) (4 pts) Illustrate how a heap delete operation would work, redrawing the tree at each step.

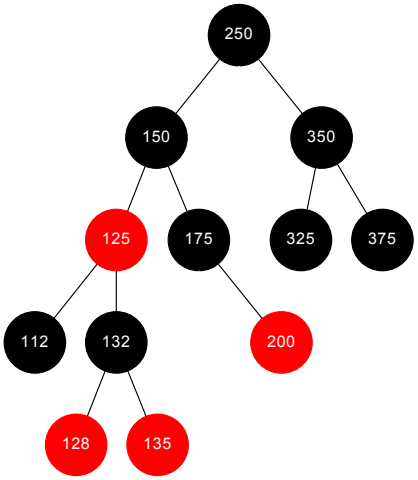

Figure 1: A Red-Black Tree

(c) (4 pts) Illustrate how a heap insert of the element 40 would work, redrawing at each step. NOTE: Start with the original heap (not with the heap resulting from problem 6b)

7. Suppose a red-black tree is implemented using the following class:

```
class RBNode {
   int data;
   RBNode left, right, parent;
   char color;
}
```

   (a) (8 pts) Write a method called `bheight`, which given the root of a red-black tree will return the black height of the tree. Assume that the color field stores a `'B'` for a black node and an `'R'` for a red node. Assume the existence of a global variable named `NIL` that is black in color and is used in place of `null` to mark the bottom of the tree. Do not include the `NIL` node in the calculation of black height.

   (b) (8 pts) Write a method called `leftRotate`, which given the a reference to an RBNode, will perform a left-rotate operation. Assume the existence of the global variable `NIL` as in the previous example. Also assume the existence of a global variable called `root` that is a reference to the root of the tree (which will be pointing to `NIL` if the tree is empty).

8. (8 pts) Suppose there are two distinct, existing hash functions named `hash1` and `hash2` that accept a string as a parameter and return an int in the range 0 to 9999. Write a function called `insert` that accepts a string to be inserted as a parameter and uses double-hashing as the collision-resolution strategy.

   Assume the existence of a class-wide variable called `table` that holds room for 10000 string variables. Don't forget that you can use the pre-existing `hash1` and `hash2` functions.

9. Consider the following excerpt of code for a container class that holds double values as its data items:

```
class MyContainer
{
   private double [] data;
   private int n;

   ...

   public MyContainer() {
      data= new double[1000];
      n= 0;
   }

   ...

   public void sort() {
      int i,j;
      double temp;
      for (i=0; i<n-1; i++)
         for (j=i+1; j<n; j++)
            if (data[i]>data[j]) {
               temp= a[i];
               a[i]= a[j];
               a[j]= temp;
            }
   }
}
```

   (a) (8 pts) Show changes you would need to make to this existing code to make `MyContainer` to be generic. You can write your answers on this sheet.

   (b) (6 pts) Write a class called `SugarGlider` that has as attributes for the name (String) and wingspan (double) of a single sugar glider. Add methods that would be required for the `SugarGlider` class to work with your generic `MyContainer` class so that the `sort()` method would sort the sugar gliders by wingspan.