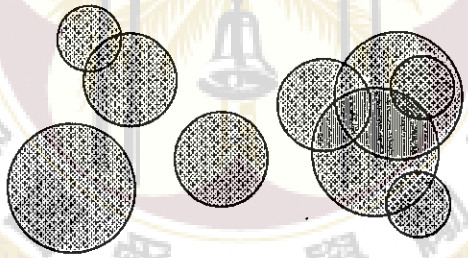
- You have to reason your answers unless told otherwise. Correct answers with weak reasons may result in deduction in grade. You get a higher score for better designs/time complexity.

1. (20%) Devise a data structure that supports insertion() and median(). The function insertion() inserts a real number (element) into the data structure, while median() returns the **median** of the elements currently in the data structure (without deleting it). Suppose that there are n elements in the data structure. For an odd n, the median is the $(n+1)/2$-th smallest element. For an even n, the median is the mean of the $(n/2)$-th and the $(n/2+1)$-th smallest elements. Describe your data structure in detail. What is the time complexity in total for n calls of insertion() plus $O(n)$ calls of median() in random order? Of course, median() won't be called twice without any insertion() in between.

2. (25%) Open-addressing Hashing
   (a) (5%) Write down a second hash function, h_2, of double hashing such that the double hashing performs exactly as linear probing.
   (b) (5%) Is the above hash function uniform? Why or why not?
   (c) (15%) Assume the hash function is uniform. Given the loading factor is α, what is the average number of comparisons for an un-successful look-up? You need to show the derivation.

3. (20%) Suppose that you have n circles on a 2-D plane. The radius and the center coordinate of each circle can be retrieved in O(1) time. A **closed region** is defined as a non-empty set of connected 2-D points, and each point is covered by at least one circle. For example, 9 circles form 4 closed regions in the picture below.



   (a) (10%) Your task is to find the number of closed regions. In the above picture, such number is 4. Describe your algorithm and data structure in detail. What is the time complexity of your algorithm?
   (b) (10%) Now we start to add more circles one-by-one to the plane. After each addition, we want to keep track of the number of closed regions. Describe an algorithm and data structure to do so. What is the time complexity of your algorithm for each addition?

4. (35%) Answer (c)~(e) by considering an AVL-tree with n elements.
   (a) (5%) In a red-black tree, let r be the number of red internal nodes, and b be the number of black internal nodes. What is the maximum of r/b?
   (b) (5%) Draw a 12-node AVL-tree (topology only) with the most height possible (no reason is needed).
   (c) (5%) How many rotations are needed during one insertion? Why?
   (d) (10%) Now we add a field *lsize* to each node in the tree, which records the number of nodes in its left subtree plus 1. What is the time complexity to update such information during one insertion?
   (e) (10%) With the *lsize* information in (d), describe how to find the k-th smallest elements in an AVL-tree. What is the time complexity?

試題隨卷繳回