CS314 Algorithms Design and Analysis
Problems in Amortized Complexity (Chapters 17, 19, 21 of CLRS 3rd edition)

Hand in the first 4 problems on or before August 19, 2016. The solution you hand in must be your own even if you discussed with others. If you do discuss with others or use other resources, please acknowledge. If you copy or let someone copy, then your answer sheet and further submissions (including tests) will NOT be evaluated. You are welcome to contact the instructor or the TA for any clarifications.

1. Exercise 17.3-3. Do you think you can make the amortized cost of both operations $O(1)$? If that is possible, how fast can you sort a list of n elements?

2. Exercise 17.3-4

3. Problem 17-2

4. Let $A[1], A[2], \ldots A[n]$ be a given sequence of n distinct numbers. The cartesian tree of the sequence is defined inductively as follows. The cartesian tree of a single element (i.e. $n = 1$) is a single node containing the element. For $n \geq 2$, the root node contains the smallest element of the sequence. Let i be the position of the smallest element. Then the left subtree (if any) of the root is the cartesian tree of $A[1], A[2], \ldots A[i-1]$ and the right subtree (if any) of the root is the cartesian tree of $A[i+1], A[i+2], \ldots A[n]$.

   Note that the cartesian tree follows the minheap property among the nodes and their children.
   Consider the following algorithm to insert a new key element x into an existing cartesian tree T that does not contain x. Simply follow the rightmost path of the tree and insert x as a right child of the leaf y in the rightmost path. Note that y is currently the parent of x. Then execute the following steps.
   while $x < y$,
   left rotate at x so that y becomes the child of x
   $y \leftarrow$ the new parent of x
   endwhile

   Convince yourself that at the end this results in a cartesian tree.

   (a) What is the worst case complexity of insertion?
   (b) Show that the amortized complexity of insertion is a constant (Hint: Try some small examples, and identify a simple potential function.)

5. Exercise 17.3-7

6. Exercise 19.4-1

7. Exercise 19.4-2

8. Exercise 21.3-3

9. Exercise 21.3-5

10. Problem 21-1