CS 473: Undergraduate Algorithms, Spring 2009
HBS 5

1. Recall that the staircase of a set of points consists of the points with no other point both above and to the right. Describe a method to maintain the staircase as new points are added to the set. Specifically, describe and analyze a data structure that stores the staircase of a set of points, and an algorithm $INSERT(x, y)$ that adds the point (x, y) to the set and returns $TRUE$ or $FALSE$ to indicate whether the staircase has changed. Your data structure should use $O(n)$ space, and your INSERT algorithm should run in $O(\log n)$ amortized time.

2. In some applications, we do not know in advance how much space we will require. So, we start the program by allocating a (dynamic) table of some fixed size. Later, as new objects are inserted, we may have to allocate a larger table and copy the previous table to it. So, we may need more than $O(1)$ time for copying. In addition, we want to keep the table size small enough, avoiding a very large table to keep only few items. One way to manage a dynamic table is by the following rules:

   (a) Double the size of the table if an item is inserted into a full table

   (b) Halve the table size if a deletion causes the table to become less than $1/4$ full

   Show that, in such a dynamic table we only need $O(1)$ amortized time, per operation.

3. Consider a stack data structure with the following operations:

   - PUSH(x): adds the element x to the top of the stack
   - POP: removes and returns the element that is currently on top of the stack (if the stack is non-empty)
   - SEARCH(x): repeatedly removes the element on top of the stack until x is found or the stack becomes empty

   What is the amortized cost of an operation?