CS 180: Data Structures, Fall 2011
Homework 3

Due by the start of class on Friday, Nov. 11

1. Write a C++ function called shuffle that takes two STL integer lists as inputs, the first as a reference parameter and the second a const reference. The goal is to alter the first list (but not the second) so that the nodes of the first list alternate with nodes of the second list.

   For example, if first contains the values 1, 2, 3, and second contains the values 4, 5, 6, at the end your first list should be 1, 4, 2, 5, 3, 6 (and the second list should be unaltered).

   Note that the lists need not be the same length, and your code should just insert without altering if necessary. For example, if the lists are 1, 2 and 4, 5, 6, 7, your first list should end as 1, 4, 2, 5, 6, 7. If the first list is 1, 2, 3, 4, and the second is 5, 6, you should end with 1, 5, 2, 6, 3, 4.

   Below is the beginning of your function; your job is to code the rest. Note that this is NOT using our list class - you must use the STL for this one, so don't forget to import list in the usual way. You must also include a small main function which you use to test your list. Please submit one cpp file with both the function and your main, for ease of testing.

   ```
   void shuffle(list<int>& first, const list<int>& second) {
   ```

2. Code one of the sorting algorithms from class in either our vector or our list class. Don't forget to actually test that it works, and submit a test file also. (Feel free to use the other functions from that class, such as insert, etc.)

3. (a) Suppose that we start with an initially empty max heap, and that the following items are inserted: 22, 28, 9, 3, 6, 33, 11, 32, 7, 19.
   Draw the heaps that results after these operations **in this order**. While I do not require you to show your work, I encourage you to do so for the purposes to partial credit and to double check your work.
   (b) Now draw the max heap that results after removeMax is called on your heap from part a. Again, I encourage you to show your work.

4. Draw the binary search tree that results after the following elements are inserted into an initially empty BST **in this order**: 16, 4, 6, 34, 2, 7, 24, 42, 1, 17

5. (a) Your classmate claims that the order in which a set of elements is inserted into a binary search tree does not matter - the same search tree results each time. Give a (small) example to show they are wrong.
   (b) Now this same classmate claims that a preorder traversal of a heap will list its keys in sorted order. Give a (small) example of a head that proves he is still wrong.

6. **Extra Credit**: Code a different sorting algorithm in the other class, Vector or List (from whatever you chose to implement in problem 2).