Lab 7

1. Devise an O(n) algorithm to accomplish this task: You are given a `List` of `Strings`. You must thin out this `List` by making sure that no `String` occurs *more than twice* in the `List`.

   For example, you are given a list like following:
   [andy, mike, andy, andy, michelle, mike]
   Your algorithm should return a list like following:
   [andy, mike, andy, michelle, mike]
   (Note that "andy" occurs more than twice, and we only output twice in that case.)

2. Devise an O(n) algorithm to accomplish this task:
   Given a none-empty string S of length n, S consists some words separated by spaces. We want to reverse every word in S.
   For example, given S = "we test coders", you algorithm is going to return a string with every word in S reversed and separated by spaces. So the result for the above example would be "ew tset sredoc".

3. Create a sorting routine based on a BST and place it in the sorting environment, distributed earlier. For this, your new class, BSTSort, should be a subclass of Sorter. Your BSTSort class can be essentially the same as the BST class given in the slides (see the folder in your labs directory for this lab), except that you will need to modify the printTree method so that it outputs values to an array (rather than printing to console).

   After you have implemented, discuss the asymptotic running time of your new sorting algorithm. Run an empirical test in the sorting environment and explain where BSTSort fits in with the other sorting routines (which algorithms is it faster than? which is it slower than?).

4. For each integer *n* = 1, 2, 3,…, 7, determine whether there exists a red-black tree having exactly *n* nodes, with *all of them black.* Fill out the chart below to tabulate the results:

Num nodes *n*	**Does there exist a red-black tree with *n* nodes, all of which are black?**
1	Yes
2	
3	
4	
5	
6	
7	

5. For each integer *n* = 1,2,3,…, 7, determine whether there exists a red-black tree having exactly n nodes, where *exactly one of the nodes is red.* Fill out the chart below to tabulate the results:

Num nodes *n*	**Does there exist a red-black tree with *n* nodes, where exactly *one* of the nodes is red?**
1	No
2	
3	
4	
5	
6	
7	