2. This causes compilation errors since insert() calls contains() and contains() needs to find whether these two items are equal. For int or string, != is a well-defined operator, but for Bottle, != is not well-defined.

3b. We cannot implement listAll() by recursion directly since we need a string variable, which is path in listAllAuxiliary(), that can keep the names of all directories of a certain plain file in a required form. Without that variable path, we may only output the upper one level directory of a plain file rather than completely output all levels directories.

4a. The time complexity of this algorithm is 0(N3) because the innermost for-loop is nested into the outer one and the outer for-loop is nested into the outermost one. The time complexity should be O (N\*N\*N)=O (N3)

4b. The time complexity of this algorithm is also O(N3) because even though the limit of j is not N but i, as i in the outermost for-loop grows, the times of assignment operations also grows linearly. That is, for outer two for-loops, the total times of operations should be 1/2N2. The time complexity is O(N2). Then, since another for-loop is nested into these two existing for-loops, the entire time complexity should be O (N2\*N)=O (N3)

5a. The worst case is that S1 and S2 are totally different Set. That is, S1 and S2 do not have any common elements. Copying S1 to result takes N steps. Then for each k in the for-loop, we would call get() and insert() once. insert() calls contains() and contains() calls find(). Since S1 and S2 have no common elements, find() will execute N times. Therefore, the total number of steps is about N+N+N\*N+N=3N+N2 and the time complexity is O(N2)

5b. The time complexity of this algorithm is NlogN. Copying all elements of S1 and S2 into the vector involves 2N steps. As indicated, the time complexity of the sorting algorithm is NlogN. Deleting the result nodes needs N – 1 comparisons and at most N – 1 calls of doErase(). For the last step, copying unique items from vector into result involves a for-loop, and for each k, the program executes the if-statement and calls insertAtTail() once. So the number of steps for this subpart is 3N. The total number of steps is about 7N+NlogN. Therefore, we know that the time complexity is O(NlogN).