Kyle Kodani

HW4

2) The insert function calls contains, which calls find, and find uses the != operator to compare data stored in the nodes. Since the data in the node is a bottle, and the bottle class doesn’t have a defined != operator, there is an error.

3b) Without the path parameter, there is no way to track how deep we are in the recursion. The path parameter remembers what recursion level the function is currently at (AKA what directory we are in).

4a) Each for loop loops for N times, and does basic tasks like addition, assignment, etc. These tasks only happen a finite number of times per loop, so each task effectively happens N times. Since there are three for loops that are nested in each other (i.e. a loop inside a loop inside a loop), the time is O(N³)

4b) The time is still O(N³). Setting the middle for loop to loop i times only reduces the number of iterations to about N/2, which still simplifies to N thus getting us O(N³).

5a) The worst case time is O(N²). The unite function calls the assignment operator which calls the copy constructor which visits N nodes. The unite function also has a for loop that runs N times. Each time through the loop the get function is called, which visits N/2 nodes, and the insert function is called, which visits N nodes (in actuality, insert calls contains which calls find which runs for N). Thus we have N+N((N/2)+N) which simplifies to N².

5b) The running time is O(N\*log(N)). This two for loops at the start of this function both visit N objects each, thus we have 2N. The sorting algorithm is called on the combination of the two sets, so its time is 2N\*log(2N). The while loop does not visit any objects, so its time is 0. The final for loop visits two objects per loop, and it loops 2N times, so its total time is 4N. Thus we have 2N+2N\*log(2N)+4N, which simplifies to N\*log(N).