2. The call to Set<Coord>::insert() causes a compilation error because the comparison operator < is not defined for the Coord class. As a result, when Set<ItemType>::findFirstAtLeast() tries to compare two Coord objects with the < operator, a compilation error occurs.

3b. Without the second string parameter, I would not have been able to keep track of the previous menu item names and how many backslashes to put in the menu name. I’d only be able to traverse the menu tree and print out individual names only.

4a. The Big O of the program in the problem is N³ because there are three layers of for loops, each iterating N times, so the operation k++ will occur N³ times.

4b. The Big O of the program is still N³ because i is incremented N times, j is incremented N(N - 1) times, and k is incremented N²(N -1) times. The highest order term still is N³.

5a. The worst case time complexity of the unite function written in the problem is O(N²) because the worst case is if result, s1, and s2 are all distinct sets with no common values. In that case, result is automatically set to s1. When insert is called, the function findFirstAtLeast() iterates at most N times, and insert is called N times, so the time complexity is N².

5b. The time complexity is O(NlogN), because it takes 2N iterations to fill the vector v, NlogN iterations to sort out v, N iterations to delete nodes in result, and 2N more iterations to enter values from v into result. The greatest use of time is the sorting algorithm.

5c. The time complexity is O(N) because the while loop can iterate at most 2N times. The two loops are not nested, and the only function called within a loop is insertBefore(), which has an O(1).