Edmund Zhi

805328871

Homework 4

2. The call to Set<Coord>::insert causes a compilation error because we are inserting an object of the Coord type, but in our insert function, we make a call to findFirstAtMost, which utilizes a comparison operator (>) in its for loop. However, we have not defined what it means for a Coord to be > another Coord. This means that we cannot make this comparison and there is a compilation error.

4b. You wouldn’t be able to pass down and keep track of the “path” string as a parameter for recursive calls with greater depth.

5a. O(N^3) because we have three layers of loops that search through N items. Our outermost for loop searches through at most N items. However, for each of those N items, we do another search through N items. But for each of those N items again, we have to do a search through N more. Therefore, we find a time complexity of N \* N \* N, leading to O(N^3).

5b. The time complexity of this function is still O(N^3) because even if the loop limit becomes i, and not N, i still has the potential to grow up to as large as N in the worst-case scenario, and we calculate Big-O based on this. Again, we have three nested for loops that all run up to N times, leading us to calculate the time complexity to be N \* N \* N, or O(N^3).

6a. Unite uses the helper functions get and insert. Get splits up the set so that it searches through either the first half or second half, making the worst case search N/2 times. Insert uses the helper function findFirstAtMost. findFirstAtMost will, at worst case, have to do N searches since it would look all the way through a list for a match. Unite then searches through each of the N elements in the set, and for each of those elements it does N/2 searches through get() and then N searches in insert. So we get N \* (N/2 + N), which leads us with a Big-O value of O(N^2).

6b. Copying all items from set 1 and set 2 into V will have a time complexity of N + N. Then, deleting all of the N nodes in set 3 will also have a complexity of N. Running sort, which uses an O(N logN) algorithm, will have a complexity of N logN. Lastly, copying all of the 2N items in V into our result will have a complexity of 2N, because the outside loop has complexity of 2N inside the loop, the function call insertBefore() has constant time, making each of the inside loop calls constant time as well. Therefore, in total, this unite function has a total complexity of 2N + N + N logN + 2N, and since we determine the complexity based off of the highest order complexity, our overall time complexity is O(N log N).

6c. Up until the while loop, every line runs in constant time. In both our while loop and our for loop, we find that there are no nested loops inside. Since each loop has worst-case complexity N, we add them together to get 2N, but since we only care about the highest degree term (without the coeffecients), we can determine that our overall time complexity is O(N).