Oo, Thant Zin (Andy)

CS 32, TA: Ling Ding

# Homework 4

2. The call: sc.insert(Coord(7, -7));

causes an error because the insert function makes use of the == comparison operator. Because Coord is a user-defined class, in order to use the insert function, we need to define an overloaded comparison operator for Coords.

3b. Given the constraints in part a, I would not have been able to solve this problem if the recursive function only took in one parameter. The reason for this is that without being able to use other STL containers, keeping track of the full path string would not be possible.

4a. The time complexity of this algorithm is O(N^3). This is because for a given N, there are three nested for loops, each iterating N times.

4b. The time complexity of this algorithm is still O(N^3). This is because in the worst case, wherein i = N – 1, the inner for loop will run approximately N times. As such, this algorithm’s time complexity is of the same order as the previous algorithm.

5a. Assuming that neither set1, set2 or result are pointers to the same set, the worst-case time complexity of this function is O(N^2). This is because, the for loop:

for (int k = 0; k < sp->size(); k++)

will run N times. Each iteration will call get() and insert(), both of which are functions with O(N) complexity. (Max number of nodes visited for get is ½ \* N, max for insert is N). As such, the time complexity is N^2.

5b. The time complexity of this algorithm is O(NlogN). The first two for loops will iterate at most N times. Given that sorting v entails O(NlogN) complexity, we can discount the first two loops as NlogN is of higher order than N. The while loop iterates N times, and doErase() is a constant time function, therefore the while loop visits ItemTypes roughly N times. Lastly, assuming v.size() == N, the last for loop iterates N times. insertBefore() is a constant time function, therefore the final for loop is also roughly O(N). The highest order within this entire function remains NlogN, therefore the time complexity is O(NlogN(.

5c. The time complexity of this function is O(N). Assuming this, set1 and set2 are unique sets, the while loop will iterate in order N. As mentioned before, insertBefore() is a constant time function, so the while loop will be of O(N) complexity. The for loop is also O(N) for the same reason. As such, the highest order within the function is N, therefore the time complexity is O(N).