2. This line of code causes a compilation error because a Coord is a user defined class and we must define a comparison operator for that class. Map<int, double> worked because they both are primitive types of data and are already defined.

3. b) There would be no way to solve the 3a with only one parameter with a recursive solution because the second parameter allowed for us to keep track of the path of the string, allowing for us to know the parent category, like actor, alien, etc. Without that second parameter, we wouldn’t be able to print out “Actor=>Alien=>Smallgon” and only the current object per line.

4. a) **O(n^3)** because the outer for loop runs n times and the two inner four loops also run n times so the time complexity would be n(n + 1)(n+ 2) but all the lower order operations can be ignored which results in just n\*n\*n. The max number of times the loops run are n times.

b) **O(n^3)** because even though a lot of operations are skipped and the function is more simple, things do not change in terms of time complexity. Also, for the second for loop, the conditions are related to the size if i, which changes. However, the greatest i can be is (n-1), but when you ignore the lower order operations, its n. So it would be n\*n\*n.

5. **O(n ^2)** because the time complexity in terms of the linked list nodes would be n(n + n + log(n) + log(n)), and all the lower order expressions can be ignored. The smaller maps size is n elements and the for loop runs n times and the get function in the for loop has to traverse through the entire structure in the worst case.