**2. Explain in a sentence or two why the call to Map<Coord, int>::insert causes at least one compilation error.**

The class Coord does not have an assignment operator, which is needed to assign the Coord key to the newly-inserted node and to check if the Coord keys match.

**3b. Why could you not solve this problem given the constraints in part a if we had only a one-parameter listAll, and you had to implement it as a recursive function?**

Needed the extra parameter variable (string *path*) to keep track of the string pathsfor the subclasses that have been traversed through. Otherwise, the void function would not have any other way of concatenating and storing the path string without this additional parameter.

**4a. What is the time complexity of this algorithm, in terms of the number of basics operations performed. Why?**

O(N3). There are 3 embedded for-loops. All 3 loops have variables starting from 0 and ending at N, with increments of 1 each iteration of the loop. All other actions in the algorithm can be considered as constants.

**4b. What is the time complexity of this algorithm? Why?**

O(N3). In this algorithm one of the embedded for-loops iterates through 0 to i number of times (where i is the iterator count of the outer for-loop), meaning it is called about N/2 times. Therefore, the overall time complexity of the algorithm should be about N\*(N/2)\*N. However, since we drop coefficients and keep only the highest order, the time complexity would still remain the same as in 4a.

**5. What is its time complexity? Why?**

O(N2). There is a for-loop that runs N times. During that loop, we call several functions (like the get function) which presumably could loop through N times again. All other coefficients and constants are dropped, resulting in the stated time complexity.