**2. Explain in a sentence or two why the call to Map<Coord, int>::insert causes at least one compilation error. (Notice that the call to Map<int, double>::insert is fine.) Don't just transcribe a compiler error message; your answer must indicate you understand the the ultimate root cause of the problem and why that is connected to the call to Map<Coord, int>::insert.**

The class Point doesn’t have an equality or assignment operator. That operator is needed to assign the “Point” key to the newly-inserted node and to check if the “Point” keys match.

**3.b We introduced the two-parameter overload of listAll. 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 the recursive function?**

We needed a variable to keep track of the string path for the subdomains (that we’ve traveled thus far), the parameter variable path in this case. The void function has no other way of concatenating and storing the path string without that extra parameter.

**4a. What is the time complexity of this algorithm, in terms of the number of basic operations (e.g., additions, assignments, comparisons) performed: Is it O(N), O(N log N), or what? Why?**

There are three for-loops, each embedded in the other. They all have variables that begin from 0 and end at N, with an increment of 1. Each operation in side these three loops and at most execute N times. Therefore the efficiency would be O(N^3) or big-O of N-cubed.

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

Basically, the middle for loop use the bounds from previous for loop which at least cut the total number of operation inside the middle for loop to half, so the total operation for this program is N\*(1/2)N\*N which is also O(N^3) or big-O of N-cubed.

**5. Assume that m1, m2, and the old value of result each have N elements. In terms of the number of linked list nodes visited during the execution of this function, what is its time complexity? Why?**

The total operation of this function depends on the for loop and the function calls which calls the find function that at most go through the whole list inside the for loop. And the get function, erase function and insert function which calls find has at most a\*N operation and three of them would be 3a\*N. And the for loop executes N times. So the time complexity is 3a\*N\*N which is O(N^2).