2. Calling Map<Coord, int>::insert will cause an error because the insert function calls several other functions including the find function. The problem is that the find function tries to compare KeyType values using the != operator but such an operator has not been defined to compare Coords, as a result an error is created.

3b. The reason a two parameter implementation of listAll() is necessary, is because the recursive function needs some way to keep track of the current instance of the function. Without such a path, there is no way to backtrack through the recursive implementation since the function is void i.e. No returns.

4a. The time complexity of the algorithm is O(N3) due to the fact that there are three nested loops running from 0 to N.

4b. The time complexity of the second algorithm is O(N3) because there are three nested loops. The outermost and innermost loops run from 0 to N still, but the central loop is only called from 0 to the outer loop limit. We disregard any of the lower order terms.

5. The time complexity of this function is approximately O(N2). There is the main for loop that iterates from 0 to N-1, and then a call to get() followed by a call to either insert() or erase() that are also looped N times. As a result, the big-O is N\*(N+N) depending on the parameters. The lower order terms are disregarded, leaving us with O(N2).