Atij Mahesh

905954423

CS32 HW4

1e) In the third test case, an iterator is used to iterate over the vector (rather than an indexing variable), but since the iterator wasn’t updated to point at the vector, the iterator became invalid. It displays undefined behavior, thus adding more than 5 elements to the vector, all initialized with unknown values.

3) The call to Set<Coord>::insert causes at least one compilation error because the class Coord does not provide a less-than operator (<) or a comparison function, which is required for the implementation of the Set class to determine the order of the elements. Since there is no way for the Set Class to determine element order, there is no way to check for uniqueness inside the Set class.

4b) The one-parameter overload of listAll only takes in a pointer to a File object and does not have a parameter for the current path. Therefore, it would not have any information about the current directory it is in, and would not be able to print out the complete pathname for each file and directory. To implement the functionality required by the problem statement, we need to keep track of the path as we traverse the directory structure recursively. The two-parameter overload of listAll, which takes in both the current path and the pointer to the File object, accomplishes this and enables recursively printing out the complete pathnames for each file and directory.

5a) The time complexity would be O(N^3). This is because there are 3 nested loops, all of which run for a maximum of N times each. There are some constant time operations inside the inner-most for loop, but these operations are of a lower degree than 3, and therefore are not involved in the final complexity calculation.

5b) The time complexity is O(N^3). Each of the 3 nested for loops still runs a maximum of N times (the middle loop runs N-1 times and the innermost runs N-2 times but only the highest degree is relevant), and thus the time complexity would still be O(N^3). Although this algorithm improves upon part A by using symmetry, this reduces the runtime by a constant coefficient (50% on average), and the loop still performs N^3 operations on N items.

6a) The worst-case time complexity is O(N^2). The worst case is when both sets have no common elements, and the function will iterate over all elements of both input sets, requiring O(N) iterations of the loop. Within each iteration of the loop, the insert function may need to traverse the entire linked list of the result set to find the correct location to insert the new element. This also requires O(N) time in the worst case. O(N) x O(N) = O(N^2).

6b) The time complexity is O(NlogN). There are multiple operations of O(N), such as copying all items into the vector, deleting result Nodes, and copying unique items from v into the result. However, the time complexity of the sort is NlogN, which is of greater magnitude than N. Therefore the complexity is O(NlogN).

6c) The time complexity is O(N). The first few operations are simple comparisons and are of constant time. In the later operations, the worst-case possibility is that both Sets are iterated through once, thus the total number of linked list nodes visited will be 2N at most. Since insertion in linked lists is constant time, the time complexity would be O(N).