Concluding on the graphs.

1. Graph - The inserting function.

For the insert function, the list structure is by far the quickest one of the three. Slowest is the sorted dictionary (using log N for each insertion, keeping the tree structure). In between the two is Dictionary. A lot faster than the sorted dictionary, but still slower that the list. The Dictionary takes a bit more time because of the hashing function.

2. Graph – The contains function.

The list is by far the slowest one in this test. It uses linear time searching for each element (breaking if found), which increases heavily with higher numbers of elements. If the list is sorted it can be searched with binary-search. Taking the time to sort the list makes it a lot faster but the cost of that sorting function is not always worth it. Dictionary and sorted dictionary are also fast. Choosing between sorted dictionary and sorted list really depends on when you need to save time. Slow insertion vs. having to sort to gain fast lookup times. The dictionary didn't fit on our graph but we included the F(x) function to illustrate the relative small time/sec even for high number of elements.

In conclusion each structure has its advantages. List has a very fast insert method, while a bit slower when having to search through it. Dictionary has advantages with fast lookup-times etc. Each structure uses rebuild functions expanding their internal arrays which is also reflected in the small deviations from the linear results.