Anosh Abraham
2/3/23

Programming Assignment 1 Discussion

For each case, print n, input array, output array (sorted data), and the number of key-comparisons. Does the number of key-comparisons agree with the theoretical values?

The number of key-comparisons for the most part agree with the theoretical values. The worst-case scenario for insertion sort is an array that is sorted in reverse order in which the number of key comparisons would be the square of the number of elements in the array (31^2=961). With the best case it will always be the length of the array minus 1, so in this case since 32 is the array length, then 31 would be the comparisons made. But with the average case, the number of key comparisons in insertion sort is also proportional to the square of the number of elements in the array, but with a lower constant of proportionality. This means that there will be some standard-deviation from the theoretical value with the key-comparisons for the average case.

Does the number of key-comparisons show O(n2) performance? That is, when the array size is increased by a factor of 10, does the number of operations (comparisons) increase by approximately a factor of 100? What is the constant factor for the O(n2) performance?

The number of key-comparisons do show O(n^2) performance. This is due to the fact that with the number of elements multiplying by a factor of 10, the comparisons grow immensely in which case it would be by a factor of 100. In the insertion sort experiment for the worst case you can see that with n=1000 elements the comparison is 996004 which is ~999^2. When you set n=10,000 elements the number of key comparisons increases by a factor of 100X to 99960004.