Nicolas Ayala
nmayala@ucsc.edu

Time complexity of each sort and some words on the constant

Bubble sort - best case O(n), worst case O(n^2); average case O(n^2)

When n>50, the constant is about 0.75

When n<50, the constant is more sporadic, mostly because some best case behaviour is more likely to occur. If we assume it is behaving with a complexity of O(n^2), the constant fluctuates from 0.5 to 1

Shell sort - best case Ω(n) O(nlog n), average case Ω(nlogn) O(n^2)

It is difficult to determine the constant, Shell sort appears to get worse than quicksort as n increases, and better than bubble sort as n increases.

Its average asymptotic behavior is probably faster than n^2 and slower than nlogn

Quick sort - best case O(nlogn), worst case O(n^2), average case O(nlogn)

When n>20, the constant is about 2

When n<20, the sort is more likely to exhibit worst case behavior. Under the assumption that it is exhibiting average case behavior, the constant can range from 2 to 2.5

Binary insertion sort - best case O(n), worst case O(n^2), average case O(n^2)

Empirical note: As far as I have seen the number of moves for binary insertion sort *exactly* equals the number of moves for bubble sort. Here's why I think this may be the case:

Claim: For both algorithms(binary insertion sort and bubble sort)

For each element A

The number of times it is swapped is equal to # of elements greater than it that are below it + # of elements lesser than it that are above it

Here's why:

I: Both of these algorithms only swap adjacent elements.

II: Both algorithms will only swap adjacent elements if the lower element is the greater of the two.

III: From **I** and **II** it follows that for each element A:

The elements greater than A yet below A must swap with A once to surpass it
The elements greater than A yet above A will not swap with A
The elements lesser than A yet below A will not swap with A
The elements lesser than A yet above A must swap with A once to subceed it
The elements equal to A will not swap with A

IV: For every element A, the other elements are exactly in one of these five sets

From **I, II, III** and **IV**, the **Claim** follows