

AP COMPUTER SCIENCE A – INSERTION SORT NOTES

The program `InsertionSort.java` sorts a list of integers using a so-called insertion sort. The method which performs the sort is on lines 32 – 53. In the table below is the state of the `int [] list` after each iteration of the `i` – loop (on lines 39 – 49). Note that the left-most column of the table contains the original, un-sorted, `list`.

For each `i` of the `i` – loop, the `a` – loop (lines 42 – 44) goes over backwards the items in `list` above `i`, and finds the topmost integer (at index `topI`) which is greater than `list[i]`. The `m` – loop (line 46) then moves each item in `list` (from `i+1` up to `topI`) down one position, and then the integer that was at index `i` is *inserted* into the position at index `topI` (hence the term *insertion sort*). Go to

https://en.wikipedia.org/wiki/Insertion_sort to see an animation of an insertion sort.

+---+---+---+---+---+---+---+---+---+---+---+										
		i		i		i		i		i
		=		=		=		=		=
		1		2		3		4		5
+---+---+---+---+---+---+---+---+---+---+---+										
	2		2		2		1		0	
	7		7		5		3		2	
	5		---		7		5		3	
	3				---		7		5	
	1				---		7		5	
	0					---	7		7	
	9						---		9	
	8							---	9	
	6								---	9
	4									---
+---+										

In any case, the `i` – loop is executed $n - 1$ times, where $n \equiv$ the length of the list. For each `i`, the combined number of iterations that the `a` – loop and the `m` – loop execute, on average, is approximately n . Thus, the total number of loop iteration is approximately $(n + 1)n = n^2 + n$, *i.e.*, the time to perform an insertion sort is $t \sim O(n^2)$, which is of the same order as it was for the selection sort.