**SUMMARY**

Intro sort aka Introspective sort is a hybrid sorting algorithm, which is a sorting algorithm that takes more than one sorting algorithm to operate, likewise intro sort uses three sorting algorithms , the following are:

* Insertion sort
* Quicksort
* Heap sort

It was intro sort was invented by David Musser in 1997. It begins with Quick sort and switches to Heap sort when the recursion depth exceeds a level based on the number of elements being sorted. Currently the fastest sorting algorithm is quicksort but it’s got a couple of edge cases when it performs really poorly – in O (n^2) time instead of the O (n\*log (n)) typical runtime. Even though a smart pivot selection makes it practically impossible to have an O(n^2) runtime, “evil” input can force the sorting to fall into this, making the running time much longer than expected. Musser had two ideas to speed up the typical quicksort algorithm:

* If the algorithm on any sub-array runs longer than expected (recursion is deeper than log(array size)) then let’s switch to heap sort which is guaranteed to finish in O(n\*log(n)) time  
  If the number of elements is small enough don’t use quicksort, simply do an insertion sort which is faster on small data sets
* These two optimizations allow introsort to outperform the “dumb” quicksort implementation on most arrays (even ~200 times on artificial “evil” arrays consisting of 100.000 elements).

When the algorithms are compared, quicksort functions better than intro sort when tail recursion is applied it takes worst case O (n2). In order to avoid its worst case time complexity we prefer Intro sort over it.

When intro sort ‘s partition size is less than 16 it switches to insertion sort, while when the input size is more then 16 and less then O(2logn) it switches to quick sort but for a large number of inputs/ data intro sort switches to heap sort. Intro sort is also an unstable sort.

Intro sort time complexity:

* Best: O(n log n)
* Average: O(n log n)
* Worst: O(n log n)