Heap Sort vs Insertion / Quick Sort

There are a few notable comparisons between the heap sort algorithm and the insertion sort and quick sort. The heap sort seems to be consistent in its time complexity, having n(n log(n)) across the board in terms of cases. Insertion sort beats heap sort only when insertion sort is used in it’s best case scenario, with a time complexity of Ω(n), and in any other case performs worse than heap sort. The quick sort algorithm matches the heap sort algorithm in it’s best and average cases, but in it’s worst case falls to the same time complexity as insertion sort. Compared to the other two, in terms of time complexity, heap sort is a clear choice for a consistently timed algorithm.

In terms of space complexity, things are a little different. Heap sort has a space complexity of O(1), while quicksort performs slightly worse at O(log(n)). However, heap sort and insertion sort have identical space complexities. That being said, if you were forced to pick between the three, insertion sort could be a good choice for less complex programs since it can outperform the heap sort algorithm. As far as consistency across use-cases goes, heap sort would probably be the better option.