# Merge Sort:

The following sorting algorithms were implemented on varying input sizes ranging from 1M to 16M numbers.

1. Merge sort using Generics: The algorithm took anywhere between 578-10950 milliseconds depending on the input size. Since Generics is used, the object to be sorted must implement the Java Comparable interface. Using Generics usually takes longer as compared to Merge sort performed on an Integer array. Time complexity: O(nlogn)
2. Integer Merge sort: This algorithm ran the fastest with runtime ranging between 132 to 2336 milliseconds. Time Complexity: O(nlogn)
3. Insertion sort: For the given input sizes, this algorithm took way too long to terminate compared to merge sort, making it the worst among the three. Time complexity: O(n^2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sorting Algorithm/input size(millisecond) | 1M | 2M | 5M | 10M | 16M |
| Merge sort(Generics) | 578 | 1133 | 2843 | 6361 | 10950 |
| integer merge sort | 132 | 255 | 663 | 1413 | 2336 |
| insertion sort | INF | INF | INF | INF | INF |