Bubble sort is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The time complexity of bubble sort is $O(n^2)$ in the worst and average cases. This is evident from the chart, where the time taken by bubble sort increases in a quadratic manner with the increase in the input size. For a small input size of 10, the time taken is relatively low, but as the input size increases to 100 and then to 10000, the time taken increases dramatically.

Merge sort is a divide - and - conquer algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves. The time complexity of merge sort is $O(n \log n)$ in all cases (worst, average, and best). The chart shows that the time taken by merge sort increases much more slowly compared to bubble sort as the input size increases. Even for a large input size of 10,000, the time taken by merge sort is significantly lower than that of bubble sort.

In conclusion, when dealing with large datasets, merge sort is far more efficient than bubble sort due to its lower time complexity. Bubble sort is only suitable for very small datasets where its simplicity might be an advantage over more complex but efficient algorithms like merge sort.