

After running the program 45 times (5 times for each threshold value) with a constant size of 12, we have determined the following number of comparisons for each sorting method:

bubbleSort

Minimum: 7

Maximum: 11

Average: 9.08 -> 9

insertionSort

Minimum: 24

Maximum: 51

Average: 33.56 -> 34

mergeSort

Minimum: 44

Maximum: 44

Average: 44

****Note: We determined that this is accurate because mergeSort will always have to do the same number of comparisons as long as the size of each list remains the same.****

quickSort

Minimum: 13

Maximum: 19

Average: 16.8 -> 17

After running our program 45 times with various threshold values for a randomly generated list of size 12, we have determined that the most efficient method for threshold values less than or equal to 12 (the size of our list) is bubbleSort, since it takes the least amount of comparisons (on average). The most inefficient sorting method for threshold values greater than 12 was mergeSort, as it always took 44 comparisons to completely sort the lists. Our sorting methods, from most efficient to least efficient, are:

1. bubbleSort (for threshold values less than or equal to the size of the list)
2. quickSort (for threshold values greater than the size of the list)
3. insertionSort
4. mergeSort

As a result, threshold values that are smaller than the size of the list are preferred for the HybridSort algorithm as it will determine which sorting methods to use depending on the list size and the T value (which will determine what the Large and Small algorithms are).