**Hands-on Exercise #4: Sorting Algorithms**

1. Add a counter to the functions insertionSort and mergeSort that counts the number of comparisons that are made. Run two functions with arrays of various sizes. At what size does the difference in the number of comparisons between significant? How does this size that the orders of these algorithms predict? Compare your analysis with the actual running times and counter as a function of the input size n = 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 8192, 16384 <time.h> and clock() function.) Use the below code to generate an array value for testing.

for (int i = 0; i < n; i++)

{

//Create an unsorted array

arr[i] = (double)(n - i);

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Comparisons | | Running Time | |
| Input n | Insertion Sort | Merge Sort | Insertion Sort | Merge Sort |
| 2 |  |  |  |  |
| 4 |  |  |  |  |
| 8 |  |  |  |  |
| 16 |  |  |  |  |
| 32 |  |  |  |  |
| 64 |  |  |  |  |
| 128 |  |  |  |  |
| 256 |  |  |  |  |
| 512 |  |  |  |  |
| 1024 |  |  |  |  |
| 2048 |  |  |  |  |
| 8192 |  |  |  |  |
| 16384 |  |  |  |  |

