Sorting algorithms time comparison

This data shown below represents the output of the run.py file after running 20 sample tests for all sorting algorithms.
The time limit has been set to 75 seconds. If exceeded, then the algorithm didn't pass the test.
N is the number of elements that need to be sorted and MAX is the biggest possible value in the array.

```
TEST #1: N = 100, MAX = 100
Py3.12 Default sort(): OK, time: 0.013s
RadixSort_b10: OK, time: 0.013s
RadixSort_b2^16: OK, time: 0.017s
MergeSort: OK, time: 0.013s
ShellSort: OK, time: 0.014s
TimSort: OK, time: 0.014s
HeapSort: OK, time: 0.013s
QuickSort: OK, time: 0.013s

TEST #2: N = 1000, MAX = 1000
Py3.12 Default sort(): OK, time: 0.012s
RadixSort_b10: OK, time: 0.011s
RadixSort_b2^16: OK, time: 0.016s
MergeSort: OK, time: 0.012s
ShellSort: OK, time: 0.010s
TimSort: OK, time: 0.014s
HeapSort: OK, time: 0.015s
QuickSort: OK, time: 0.011s

TEST #3: N = 10000, MAX = 10000
Py3.12 Default sort(): OK, time: 0.014s
RadixSort_b10: OK, time: 0.018s
RadixSort_b2^16: OK, time: 0.016s
MergeSort: OK, time: 0.029s
ShellSort: OK, time: 0.028s
TimSort: OK, time: 0.024s
HeapSort: OK, time: 0.025s
QuickSort: OK, time: 0.019s

TEST #4: N = 100000, MAX = 100000
Py3.12 Default sort(): OK, time: 0.036s
RadixSort_b10: OK, time: 0.123s
RadixSort_b2^16: OK, time: 0.064s
MergeSort: OK, time: 0.182s
ShellSort: OK, time: 0.260s
TimSort: OK, time: 0.160s
HeapSort: OK, time: 0.215s
QuickSort: OK, time: 0.117s
```

```
TEST #5: N = 1000000, MAX = 100000
Py3.12 Default sort(): OK, time: 0.266s
RadixSort_b10: OK, time: 1.620s
RadixSort_b2^16: OK, time: 0.631s
MergeSort: OK, time: 2.128s
ShellSort: OK, time: 4.543s
TimSort: OK, time: 2.035s
HeapSort: OK, time: 3.016s
QuickSort: OK, time: 1.460s

TEST #6: N = 500000, MAX = 823457266
Py3.12 Default sort(): OK, time: 0.153s
RadixSort_b10: OK, time: 0.906s
RadixSort_b2^16: OK, time: 0.314s
MergeSort: OK, time: 1.006s
ShellSort: OK, time: 1.712s
TimSort: OK, time: 0.879s
HeapSort: OK, time: 1.298s
QuickSort: OK, time: 0.616s

TEST #7: N = 100000, MAX = 1000000000
Py3.12 Default sort(): OK, time: 0.041s
RadixSort_b10: OK, time: 0.191s
RadixSort_b2^16: OK, time: 0.065s
MergeSort: OK, time: 0.182s
ShellSort: OK, time: 0.264s
TimSort: OK, time: 0.165s
HeapSort: OK, time: 0.222s
QuickSort: OK, time: 0.119s

TEST #8: N = 1000000, MAX = 1000000000
Py3.12 Default sort(): OK, time: 0.309s
RadixSort_b10: OK, time: 2.843s
RadixSort_b2^16: OK, time: 0.782s
MergeSort: OK, time: 2.251s
ShellSort: OK, time: 4.662s
TimSort: OK, time: 2.018s
HeapSort: OK, time: 3.109s
QuickSort: OK, time: 1.334s

TEST #9: N = 10000000, MAX = 1000000000
Py3.12 Default sort(): OK, time: 3.533s
RadixSort_b10: OK, time: 42.918s
RadixSort_b2^16: OK, time: 9.718s
MergeSort: OK, time: 31.041s
ShellSort: FAIL, time: 75.368s
TimSort: OK, time: 30.479s
HeapSort: OK, time: 51.823s
QuickSort: OK, time: 21.164s
```

```
TEST #10: N = 5000000, MAX = 10000
Py3.12 Default sort(): OK, time: 1.294s
RadixSort_b10: OK, time: 8.747s
RadixSort_b2^16: OK, time: 1.738s
MergeSort: OK, time: 14.200s
ShellSort: OK, time: 33.351s
TimSort: OK, time: 12.850s
HeapSort: OK, time: 22.672s
QuickSort: OK, time: 34.168s

TEST #11: N = 8000000, MAX = 256
Py3.12 Default sort(): OK, time: 1.420s
RadixSort_b10: OK, time: 4.296s
RadixSort_b2^16: OK, time: 1.961s
MergeSort: OK, time: 16.938s
ShellSort: OK, time: 22.735s
TimSort: OK, time: 14.884s
HeapSort: OK, time: 22.603s
QuickSort: FAIL, time: 75.022s

TEST #12: N = 10000000, MAX = 65536
Py3.12 Default sort(): OK, time: 3.023s
RadixSort_b10: OK, time: 18.244s
RadixSort_b2^16: OK, time: 8.458s
MergeSort: OK, time: 30.365s
ShellSort: FAIL, time: 75.377s
TimSort: OK, time: 28.815s
HeapSort: OK, time: 52.210s
QuickSort: OK, time: 38.068s

TEST #13: N = 15000000, MAX = 100000
Py3.12 Default sort(): OK, time: 4.746s
RadixSort_b10: OK, time: 35.113s
RadixSort_b2^16: OK, time: 13.430s
MergeSort: OK, time: 47.482s
ShellSort: FAIL, time: 75.572s
TimSort: OK, time: 44.931s
HeapSort: FAIL, time: 75.580s
QuickSort: OK, time: 58.169s

TEST #14: N = 20000000, MAX = 1000
Py3.12 Default sort(): OK, time: 4.395s
RadixSort_b10: OK, time: 24.281s
RadixSort_b2^16: OK, time: 6.791s
MergeSort: OK, time: 60.061s
ShellSort: FAIL, time: 75.548s
TimSort: OK, time: 55.316s
HeapSort: FAIL, time: 75.527s
QuickSort: FAIL, time: 75.210s
```

```
TEST #15: N = 5000000, MAX = 100000000
Py3.12 Default sort(): OK, time: 1.636s
RadixSort_b10: OK, time: 17.731s
RadixSort_b2^16: OK, time: 4.537s
MergeSort: OK, time: 14.094s
ShellSort: OK, time: 39.797s
TimSort: OK, time: 13.537s
HeapSort: OK, time: 23.518s
QuickSort: OK, time: 9.392s

TEST #16: N = 10000000, MAX = 2147483647
Py3.12 Default sort(): OK, time: 4.260s
RadixSort_b10: OK, time: 45.963s
RadixSort_b2^16: OK, time: 10.457s
MergeSort: OK, time: 31.072s
ShellSort: FAIL, time: 75.364s
TimSort: OK, time: 29.689s
HeapSort: OK, time: 52.968s
QuickSort: OK, time: 20.220s

TEST #17: N = 12000000, MAX = 1000
Py3.12 Default sort(): OK, time: 2.596s
RadixSort_b10: OK, time: 14.629s
RadixSort_b2^16: OK, time: 3.931s
MergeSort: OK, time: 33.351s
ShellSort: OK, time: 74.287s
TimSort: OK, time: 31.482s
HeapSort: OK, time: 48.522s
QuickSort: FAIL, time: 75.116s

TEST #18: N = 15000000, MAX = 16777216
Py3.12 Default sort(): OK, time: 5.531s
RadixSort_b10: OK, time: 52.191s
RadixSort_b2^16: OK, time: 13.903s
MergeSort: OK, time: 49.051s
ShellSort: FAIL, time: 75.569s
TimSort: OK, time: 45.720s
HeapSort: FAIL, time: 75.569s
QuickSort: OK, time: 32.476s

TEST #19: N = 50000, MAX = 50000000000
Py3.12 Default sort(): OK, time: 0.025s
RadixSort_b10: OK, time: 0.120s
RadixSort_b2^16: OK, time: 0.060s
MergeSort: OK, time: 0.097s
ShellSort: OK, time: 0.137s
TimSort: OK, time: 0.086s
HeapSort: OK, time: 0.113s
QuickSort: OK, time: 0.069s
```

```
TEST #20: N = 100000000, MAX = 100
Py3.12 Default sort(): OK, time: 16.295s
RadixSort_b10: OK, time: 50.504s
RadixSort_b2^16: OK, time: 23.147s
MergeSort: FAIL, time: 75.089s
ShellSort: FAIL, time: 75.083s
TimSort: FAIL, time: 75.084s
HeapSort: FAIL, time: 75.079s
QuickSort: FAIL, time: 75.078s
```