Dalton North
Professor Devanny
CSCI 140

Code:
https://github.com/dinorth409/algsEC/commit/8673c8ee005cf2a1c37b827144a67f5f124b9eaf

Question 1

Time to sort in seconds
Base 10

| n (list size) | Insertion | Merge |
|---|---|---|
| 5 | 0.004656 | 0.017914 |
| 20 | 0.006985 | 0.021945 |
| 50 | 0.031698 | 0.072089 |
| 100 | 0.063283 | 0.098059 |
| 150 | 0.153211 | 0.143906 |
| 180 | 0.297182 | 0.278221 |
| 200 | 0.324577 | 0.281945 |

Base 2

| n (list size) | Insertion | Merge |
|---|---|---|
| 5 | 0.001791 | 0.004746 |
| 20 | 0.004101 | 0.016433 |
| 50 | 0.023772 | 0.047927 |
| 100 | 0.05892 | 0.095511 |
| 150 | 0.141494 | 0.210967 |
| 180 | 0.282809 | 0.32656 |
| 200 | 0.280879 | 0.134274 |

Base 16

| n (list size) | Insertion | Merge |
|---|---|---|

| | | |
|---|---|---|
| 5 | 0.001402 | 0.004561 |
| 20 | 0.003946 | 0.020273 |
| 50 | 0.021545 | 0.069165 |
| 100 | 0.059105 | 0.088063 |
| 150 | 0.142107 | 0.140169 |
| 180 | 0.290824 | 0.148606 |
| 200 | 0.259211 | 0.056328 |

The program creates a list of random integers. It then creates copies of it in base 2 and 16. Finally, the two different sorts are tested on all three bases with varying list sizes. Insertion sort is consistently faster until n=180. Here, the base 2's insertion sort is still faster; however, merge sort beats it in bases 10 and 16. When n=200, merge sort is consistently faster for all bases. Overall, base 10 is the slowest on both insertion and merge sort, while base 16 is the fastest on both.

Question 2

Time of radix sort in seconds

| n (list size) | Base 2 | Base 10 | Base 16 |
|---|---|---|---|
| 5 | 0.00748 | 0.00997 | 0.005335 |
| 15 | 0.014521 | 0.030935 | 0.008749 |
| 20 | 0.016527 | 0.013558 | 0.007222 |
| 35 | 0.031502 | 0.023172 | 0.012437 |
| 50 | 0.036029 | 0.020553 | 0.011355 |

Base 16 is the fastest when sorting with radix sort. The second fastest is the decimal base, and binary comes in last.