|  |  |  |
| --- | --- | --- |
| **Selection Sort** | | |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 499500 | 0.14731240272521973 |
| **2,000 (observed)** | 1999000 | 0.549964189529419 |
| **4,000 (observed)** | 7998000 | 1.806389570236206 |
| **8,000 (observed)** | 31996000 | 6.967347860336304 |
| **16,000 (observed)** | 127992000 | 26.559900283813477 |
| **32,000 (observed)** | 511984000 | 125.61976099014282 |
| **100,000 (estimated)** | 4999950000 | 1470 |
| **500,000 (estimated)** | 124999750000 | 36750 |
| **1,000,000 (estimated)** | 499999500000 | 147000 |
| **10,000,000 (estimated)** | 49999995000000 | 14700000 |

|  |  |  |
| --- | --- | --- |
| **Insertion Sort** | | |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 246998 | 0.0601048469543457 |
| **2,000 (observed)** | 1016731 | 0.302105188369751 |
| **4,000 (observed)** | 3991281 | 1.4224083423614502 |
| **8,000 (observed)** | 16104212 | 5.027310132980347 |
| **16,000 (observed)** | 64651468 | 24.76052451133728 |
| **32,000 (observed)** | 257475139 | 83.68932676315308 |
| **100,000 (estimated)** | 2499975000 | 261.53 |
| **500,000 (estimated)** | 62499875000 | 1307.65625 |
| **1,000,000 (estimated)** | 249999750000 | 2615.3125 |
| **10,000,000 (estimated)** | 24999997500000 | 26153.125 |

1. Which sort do you think is better? Why?

Insertion sort is better because on average, it takes less time compared to selection sort to process the same number of items, especially when the list size is large. It also does less comparisons to get the same results.

1. Which sort is better when sorting a list that is already sorted (or mostly sorted)? Why?

Insertion sort because it is an O(N) operation while selection sort is a O(N2) operation when a list is sorted/mostly sorted.

1. You probably found that insertion sort had about half as many comparisons as selection sort. Why? Why are the times for insertion sort not half what they are for selection sort? (For part of the answer, think about what insertion sort has to do more of compared to selection sort.)

Selection sort compares all numbers in the “unsorted part” to find the minimum of that portion. Insertion sort takes 1 number from the “unsorted part” and compares it to each number in the “sorted part” and stops when it finds the correct spot to place in the “sorted part.” On average, the number would be directly in the center of the “sorted part” meaning that it does half of the comparisons that selection sort had to do. Because insertion sort stops when it finds the right place in the “sorted part,” it could stop after 1 comparison or possibly until it reaches the last comparison. Therefore if some of the numbers are already ordered, it would reduce the time.