|  |  |  |
| --- | --- | --- |
| **Selection Sort** | | |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 499500 | 0.0522 |
| **2,000 (observed)** | 1999000 | 0.207 |
| **4,000 (observed)** | 7998000 | 0.796 |
| **8,000 (observed)** | 31996000 | 3.52 |
| **16,000 (observed)** | 127992000 | 14.4 |
| **32,000 (observed)** | 511984000 | 65.9 |
| **100,000 (estimated)** | 2047936000 | 263.6 |
| **500,000 (estimated)** | 8191744000 | 1054.4 |
| **1,000,000 (estimated)** | 32766976000 | 4217.6 |
| **10,000,000 (estimated)** | 131067904000 | 16870.4 |

|  |  |  |
| --- | --- | --- |
| **Insertion Sort** | | |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 247991 | 0.0611 |
| **2,000 (observed)** | 1006022 | 0.205 |
| **4,000 (observed)** | 4026100 | 0.890 |
| **8,000 (observed)** | 1611526 | 3.70 |
| **16,000 (observed)** | 64334128 | 15.8 |
| **32,000 (observed)** | 256474063 | 64.6 |
| **100,000 (estimated)** | 1025896252 | 258.4 |
| **500,000 (estimated)** | 4103585008 | 1033.6 |
| **1,000,000 (estimated)** | 16414340032 | 4134.4 |
| **10,000,000 (estimated)** | 65657360128 | 16537.6 |

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

**I would think that insertion sort is better. It has less comp per list, which in cases of slow clock cycles, or slow processors, I believe would change the time taken by a lot when you get to much larger values of n.**

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

**Insertion sort would work best when a list is already sorted, as it doesn’t have much work to do when comparing values**.

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.)

**Insertion sort has half the comparisons because it’s reducing to total unordered list size by -1 with each iteration, so as we move along, we have to check less and less values. Insertion sort has about the same run time as selection sort, however. This is due to insertion sort changing the order of the list multiple times per pass instead of the one time that selection sort does it.**