Lab 6

|  |  |  |
| --- | --- | --- |
|  | **Selection Sort** |  |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 499500 | 0.16413140296936035 |
| **2,000 (observed)** | 1999000 | 0.700908899307251 |
| **4,000 (observed)** | 7998000 | 2.3051371574401855 |
| **8,000 (observed)** | 31996000 | 10.019314289093018 |
| **16,000 (observed)** |  | 31.289352655410767 |
| **32,000 (observed)** |  | 122.30750870704651 |
| **100,000 (estimated)** |  |  |
| **500,000 (estimated)** |  |  |
| **1,000,000 (estimated)** |  |  |
| **10,000,000 (estimated)** |  |  |

|  |  |  |
| --- | --- | --- |
|  | **Insertion Sort** |  |
| **List Size** | **Comparisons** | **Time (seconds)** |
| **1,000 (observed)** | 246992 | 0.386462926864624 |
| **2,000 (observed)** | 1016724 | 1.0184295177459717 |
| **4,000 (observed)** | 3991272 | 3.423034906387329 |
| **8,000 (observed)** | 16104203 | 11.15290355682373 |
| **16,000 (observed)** | 64651458 | 48.071587800979614 |
| **32,000 (observed)** |  | 210.58257603645325 |
| **100,000 (estimated)** |  |  |
| **500,000 (estimated)** |  |  |
| **1,000,000 (estimated)** |  |  |
| **10,000,000 (estimated)** |  |  |

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

I think selection sort is better because it seems faster according to the table.

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

Insertion sort is better because you can continue comparing elements without changing much because the list is presumably already sorted. Insertion sort runs O(n) when the list is sorted while Selection sort still runs O(n^2) when a list is 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.)