| **Sorting Method** | **Array Size** | **Sorting Option (ascending, descending, etc)** | **Runtime Results** |
| --- | --- | --- | --- |
| Bubble Sort | 1000 | Ascending | 30 |
| Bubble Sort | 500 | Ascending | 19 |
| Bubble Sort | 250 | Ascending | 7 |
| Bubble Sort | 100 | Ascending | 3 |
| Bubble Sort | 1000 | Descending | 1912.333 |
| Bubble Sort | 1000 | Random | 2412.666 |
| Bubble Sort | 1000 | Nearly sorted | 1811.333 |
| Selection Sort | 1000 | Ascending | 1223 |
| Selection Sort | 500 | Ascending | 429.666 |
| Selection Sort | 250 | Ascending | 385 |
| Selection Sort | 100 | Ascending | 116 |
| Selection Sort | 1000 | Descending | 1472.666 |
| Selection Sort | 1000 | Random | 678 |
| Selection Sort | 1000 | Nearly sorted | 1357.333 |
| Insertion Sort | 1000 | Ascending | 36.3333 |
| Insertion Sort | 500 | Ascending | 19.333 |
| Insertion Sort | 250 | Ascending | 9 |
| Insertion Sort | 100 | Ascending | 3 |
| Insertion Sort | 1000 | Descending | 2282.333 |
| Insertion Sort | 1000 | Random | 623.333 |
| Insertion Sort | 1000 | Nearly sorted | 481.666 |

**Discussion:**

As can be seen through these results, there are clear use cases for each of the three sorting algorithms. On average, bubble sort is the best algorithm for an ascending array and is the worst option for random or nearly sorted arrays. On average, selection sort is the best option for a descending array and the worst option for an ascending array. On average, insertion sort is the best option for a random or nearly sorted array and is the worst option for a descending array. We can also see that as the array size approaches zero, the difference between the runtime for different algorithms becomes negligible.