Selection Sort

Insertion Sort

**Ascending:** O(n2)

**Prediction:** 994.8 s.

**Descending:** O(n2)

**Prediction:** 995.1 s.

**Random:** O(n2)

**Prediction:** 995.1 s.

If you add these prediction points to each graph, it will look like an O(n2) line. As seen in the data,

Time rises too much as the input data grows in all of them.

\*\*Note that there is an error since arrays are different and the time complexity for all does vary

|  |  |
| --- | --- |
| Ascending | |
| Size | Time |
| 550000 | 0.002455 |
| 600000 | 0.002698 |
| 700000 | 0.003131 |
| 750000 | 0.003357 |
| 800000 | 0.003659 |
| 850000 | 0.003925 |
| 900000 | 0.004212 |
| 950000 | 0.004435 |
| Descending | |
| Size | Time |
| 55000 | 3.68798 |
| 60000 | 4.53492 |
| 70000 | 6.12165 |
| 75000 | 7.04316 |
| 80000 | 8.24347 |
| 85000 | 9.76839 |
| 90000 | 10.5835 |
| 95000 | 11.828 |
| Random | |
| Size | Time |
| 55000 | 1.97923 |
| 60000 | 2.23464 |
| 70000 | 3.08171 |
| 75000 | 3.30119 |
| 80000 | 4.02948 |
| 85000 | 4.24856 |
| 90000 | 5.04435 |
| 95000 | 5.94276 |

**Ascending:** O(n) Time grows linearly as data grows.

**Prediction:** 0.0498 s. If added to the graph it will still behave like an O(n) graph

**Descending:** O(n2)

**Prediction:** 1992.1171 s.

**Random:** O(n2)

**Prediction:** 996.1535 s.

For both random and descending each point grows a lot for each data size, this, making the graph n2

Both predictions if added to the graph still behaves like a n2 graph.

\*\*Note that there is an error since arrays are different and the time complexity for all does vary

Bubble Sort



**Ascending:** O(n). Time grows linearly as data grows.

**Prediction:** 0.0297 s. If this is plotted onto the graph, it fits the linear graph, making this prediction close to being right.

**Descending:** O(n2).

**Prediction:** 5977.649 s.

**Random:** O(n2).

**Prediction:** 6974.239 s.

For both random and descending each point grows a lot for each data size, this, making the graph n2

Both predictions if added to the graph still behaves like a n2 graph.

\*\*Note that there is an error since arrays are different and the time complexity for all does vary

Merge Sort

|  |  |
| --- | --- |
| Ascending | |
| Size | Time |
| 550000 | 0.068934 |
| 600000 | 0.07471 |
| 700000 | 0.082902 |
| 750000 | 0.087427 |
| 800000 | 0.092063 |
| 850000 | 0.095754 |
| 900000 | 0.103916 |
| 950000 | 0.10845 |
| Descending | |
| Size | Time |
| 550000 | 0.064995 |
| 600000 | 0.068479 |
| 700000 | 0.078996 |
| 750000 | 0.085634 |
| 800000 | 0.093978 |
| 850000 | 0.09731 |
| 900000 | 0.100324 |
| 950000 | 0.108203 |
| Random | |
| Size | Time |
| 550000 | 0.120834 |
| 600000 | 0.12893 |
| 700000 | 0.153066 |
| 750000 | 0.163133 |
| 800000 | 0.17068 |
| 850000 | 0.18753 |
| 900000 | 0.194689 |
| 950000 | 0.206015 |

**Ascending:** O(nlogn). Since every time the list need to get divided and then merged back together it demonstrates an O(N\*log(N)) growth.

**Prediction:** 1.02 s. If this point is added to the ascending graph it will fit similar to a linear graph.

**Descending:** O(nlogn) Since every time the list need to get divided and then merged back together it demonstrates an O(N\*log(N)) growth.

**Prediction:** 1.003 s. If this point is added to the descending graph it will fit similar to a linear graph.

**Random:** O(nlogn) Since every time the list need to get divided and then merged back together it demonstrates an O(N\*log(N)) growth.

**Prediction:** 2.001 s. Higher than ascending or descending since the time to calculate greater or lower values is greater than when pre-sorted.

\*\*Note that there is an error since arrays are different and the time complexity for all does vary

Quick Sort

|  |  |
| --- | --- |
| Ascending | |
| Size | Time |
| 550000 | 0.000015 |
| 600000 | 0.000016 |
| 700000 | 0.000021 |
| 750000 | 0.000023 |
| 800000 | 0.000024 |
| 850000 | 0.000025 |
| 900000 | 0.000026 |
| 950000 | 0.000027 |
| Descending | |
| Size | Time |
| 550000 | 0.000001 |
| 600000 | 0.000002 |
| 700000 | 0.000002 |
| 750000 | 0.000003 |
| 800000 | 0.000003 |
| 850000 | 0.000003 |
| 900000 | 0.000003 |
| 950000 | 0.000003 |
| Random | |
| Size | Time |
| 550000 | 0.000001 |
| 600000 | 0.000002 |
| 700000 | 0.000002 |
| 750000 | 0.000003 |
| 800000 | 0.000003 |
| 850000 | 1.80E-06 |
| 900000 | 0.000003 |
| 950000 | 0.000002 |

**Ascending:** O(nlogn)

**Prediction:** 0.000298 s.

**Descending:** O(n2)

**Prediction:** 0.00049 s.

**Random:** O(nlogn)

**Prediction:** 0.0000303 s.

Ascending and descending behave very similar to a linear line which is why they are both quicker than random. Random behaves like an n2 line. If the prediction points are added to the graph it will look more like it.

\*\*Note that there is an error since arrays are different and the time complexity for all does vary