15 integers in random order

| Quick Sort | Heap Sort | Radix Sort | Count Sort |
| --- | --- | --- | --- |
| 6 microsecs | 8 microsecs | 48 microsecs | 63 microsecs |

***\*all measurements further down are in microsecs***

1000 integers in reverse order

| Count Sort | Radix Sort | Heap Sort | Quick Sort |
| --- | --- | --- | --- |
| 604 | 1174 | 1280 | 18170 |

1000 integers 80% sorted (every 5 has +5)

| Count Sort | Heap Sort | Radix Sort | Quick Sort |
| --- | --- | --- | --- |
| 610 | 1422 | 1553 | 22880 |

1000 integers in order

| Count Sort | Radix Sort | Heap Sort | Quick Sort |
| --- | --- | --- | --- |
| 620 | 1181 | 1449 | 28653 |

10000 integers in random order

| Count Sort | Quick Sort | Radix Sort | Heap Sort |
| --- | --- | --- | --- |
| 5802 | 6782 | 18847 | 19061 |

Count Sort was almost always superior when sorting integers. Quick sort only beat it when there was such a small amount of items. Quick sort was also very poor unless there was a very large amount of data, or a very small amount. Radix and Heap were always similar.

15 floats in random order

| Quick Sort | Heap Sort | Radix Sort | Count Sort |
| --- | --- | --- | --- |
| 7 microsecs | 9 microsecs | 251 microsecs | 421 microsecs |

***\*all measurements further down are in microsecs***

1000 floats in reverse order

| Count Sort | Radix Sort | Heap Sort | Quick Sort |
| --- | --- | --- | --- |
| 624 | 1347 | 4772 | 19583 |

1000 floats 80% sorted (every 5 has +5)

| Count Sort | Heap Sort | Radix Sort | Quick Sort |
| --- | --- | --- | --- |
| 627 | 1553 | 4743 | 24298 |

1000 floats in order

| Count Sort | Heap Sort | Radix Sort | Quick Sort |
| --- | --- | --- | --- |
| 634 | 1546 | 4726 | 30296 |

10000 floats in random order

| Count Sort | Quick Sort | Heap Sort | Radix Sort |
| --- | --- | --- | --- |
| 5908 | 7148 | 20501 | 45470 |

Count Sort was almost always superior when sorting floats as well. Quick sort only beat it when there was such a small amount of items. Quick sort was also very poor unless there was a very large amount of data, or a very small amount. Count performed well, regardless. Radix and heap were always similar.

Was having trouble getting strings to work properly. But I assume they will follow the above patterns as integers were VERY identical to floats.

15 strings in random order

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

***\*all measurements further down are in microsecs***

1000 strings in reverse order

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

1000 strings 80% sorted (every 5 has +5)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

1000 strings in order

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

10000 strings in random order

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |