Result of array size : 12020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Insertion Sort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 12019 | 72234190 | 36054 | 35188557 |
| movement | 0 | 72234190 | 24037 | 35176547 |
| Total time | 1ms | 178ms | 3ms | 92ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SelectionSort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 72234190 | 72234190 | 72234190 | 72234190 |
| movement | 0 | 6010 | 1 | 12013 |
| Total time | 128ms | 163ms | 133ms | 138ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quick Sort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 139891 | 148629 | 139891 | 184185 |
| movement | 75405 | 84025 | 75405 | 93875 |
| Total time | 9ms | 11ms | 10ms | 105ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Merge Sort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 84424 | 79492 | 96440 | 147978 |
| movement | 163916 | 163916 | 163916 | 163916 |
| Total time | 48ms | 60ms | 62ms | 61ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Heap Sort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 221993 | 188801 | 221961 | 205397 |
| movement | 161735 | 143039 | 161708 | 152902 |
| Total time | 17ms | 18ms | 19ms | 19ms |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Radix Sort | InOrder | ReverseOrder | AlmostOrder | Random |
| comparison | 0 | 0 | 0 | 0 |
| movement | 60100 | 60100 | 60100 | 60100 |
| Total time | 1ms | 3ms | 3ms | 2ms |

Summary

1. Insertion Sort is very efficient only on sorted array but in reverse order, it gives worst since it has to swap every element reversely in array.
2. Selection Sort is not efficient in most case because it has to compare O(n^2) times in any cases.
3. Quick Sort is very efficient in every case except random order, it is little bit slower than merge sort because efficiency of quicksort relies on how pivot value is set.
4. Merge sort gives average efficiency on every case because it really doesn’t matter how complex orders are.
5. Heap sort performs equally well in every case.
6. Overall, Radix sort is the fastest sorting algorithm since it is sorted by digit . But the problem is that it takes more space.