Quick Sort(n=10000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Sorted** | **Almost Sorted** |
| First Element | 0.000632 | 0.035284 | 0.004165 |
| Random Element | 0.000781 | 0.000389 | 0.000621 |
| Median pivot | 0.000915 | 0.021735 | 0.002758 |
| Median(n/4,m ,3n/4) | 0.000721 | 0.000264 | 0.000379 |

Quick Sort(n=100000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Sorted** | **Almost Sorted** |
| First Element | 0.007127 | 3.291071 | 0.037925 |
| Random Element | 0.023017 | 0.012965 | 0.005187 |
| Median pivot | 0.007026 | 1.842106 | 0.040932 |
| Median(n/4,m ,3n/4) | 0.008194 | 0.001291 | 0.003189 |

Quick Sort(n=1000000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Sorted** | **Almost Sorted** |
| First Element | 1.765120 | 1.021324 | 1.512045 |
| Random Element | 0.317668 | 0.072174 | 0.103721 |
| Median pivot | 0.628321 | 0.134209 | 0.429011 |
| Median(n/4,m ,3n/4) | 0.073091 | 0.037325 | 0.052104 |

Quick Sort(n=10000000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Sorted** | **Almost Sorted** |
| First Element | 3.543036 | 2.498317 | 1.820122 |
| Random Element | 1.920437 | 0.720943 | 0.820158 |
| Median pivot | 2.956301 | 1.562903 | 1.683991 |
| Median(n/4,m,3n/4) | 1.041664 | 0.341921 | 0.508549 |

Merge Sort

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivot\_type** | **Random** | **Sorted** | **Almost Sorted** |
| **n = 10000** | 0.000879 | 0.000161 | 0.000297 |
| **n = 100000** | 0.024716 | 0.002549 | 0.004216 |
| **n = 1000000** | 0.281773 | 0.061943 | 0.083635 |
| **n = 10000000** | 3.614371 | 0.415942 | 0.721924 |

What is your observation? Which sort is faster and why?

In merge sort, if elements are already in sorted order, then merge sort executes faster than random and almost sorted orders.

Quick sort is faster when median (n/4, middle ,3n/4) pivot element is taken.