**Quick Sort(n=10000)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivot\_type** | **Random** | **Shorted** | **Almost Sorted** |
| **First(Piv-1)** | **0.001615s** | **0.086694s** | **0.057795s** |
| **Random(Piv-2)** | **0.001504s** | **0.001100s** | **0.001093s** |
| **Median(Piv-3)** | **0.001187s** | **0.000589s** | **0.048121s** |
| **Median(Piv-4)** | **0.001391s** | **0.000638s** | **0.000632s** |

**Quick Sort(n=100000)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivot\_type** | **Random** | **Shorted** | **Almost Sorted** |
| **First(Piv-1)** | **0.012841s** | **2.941747s** | **2.371254s** |
| **Random(Piv-2)** | **0.012810s** | **0.008257s** | **0.008503s** |
| **Median(Piv-3)** | **0.015619s** | **0.010352s** | **2.386888s** |
| **Median(Piv-4)** | **0.012661s** | **0.008078s** | **0.008330s** |

**Quick Sort(n=1000000)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Shorted** | **Almost Sorted** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Quick Sort(n=10000000)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Shorted** | **Almost Sorted** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Marge Sort**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pivote\_type** | **Random** | **Shorted** | **Almost Sorted** |
| **n = 10000** |  |  |  |
| **n = 100000** |  |  |  |
| **n = 1000000** |  |  |  |
| **n = 10000000** |  |  |  |

**What is your observation? Which short is faster and why?**

**Merge sort is faster, quick sort complexity is also similar to merge but in case of sorted input it takes more time to sort than merge sort.**

**In quick sort, taking the median as pivot will take lower time in execution.**

**Note: I wasn’t able to write time for remaining ones because of system incompatibility.**