Program Structures and Algorithms

Spring 2023(SEC –08)

NAME: Priyal Vimal Gudhka

NUID: 002747680

# Assignment-6 Hits as time predictor

**Task:** In this assignment, your task is to determine--for sorting algorithms--what is the best predictor of total execution time: comparisons, swaps/copies, hits (array accesses), or something else.

**Conclusion:** Executing code for Merge Sort, Quick Sort Dual Pivot & Heap sort for different array lengths and running the experiments for 20 runs it is evident that the best predictor of execution time is number of compares. To support this conclusion, Log-Log graphs have been plotted between Number of Array Elements and Time Taken to Sort & Number of Elements and Number of Compares. Code for Merge Sort, Quick Sort Dual Pivot & Heap sort have been executed twice keeping instrumentation as true for the 1st run and instrumentation as false for 2nd run. Additionally, graphs have been plotted for the same to check the execution time.

**Evidence to Support Conclusion:** Various graphs are plotted for different array lengths that compares the time taken to sort the array with respective to the Number of Elements and additional parameters like Number of Compares, Number of Swaps and Number of Hits have been calculated.

* **Tabular & Graphical representation for Quick Sort Dual Pivot with Instrumentation as True**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **QuickSortDualPivot Results with Instrumentation as true** | | | | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Compares** | **Number of Swaps** | **Number of Hits** | **Number of**  **Runs** |
| 10000 | 41.7 | 363858828 | 272955 | 365266504 | 20 |
| 20000 | 133.85 | 2037941318 | 567648 | 2040844058 | 20 |
| 25000 | 221.75 | 2456305480 | 703297 | 2459908681 | 20 |
| 30000 | 248.65 | 5075764520 | 864292 | 5080170121 | 20 |
| 35000 | 583.2 | 6351302947 | 1002682 | 6356420528 | 20 |
| 40000 | 744.4 | 9229753484 | 1144258 | 9235595498 | 20 |
| 60000 | 906.4 | 21445373162 | 1720327 | 21454152752 | 20 |
| 65000 | 872.55 | 17707570888 | 1892881 | 17717200169 | 20 |

* **Below are the log-log graphs plotted with respect to array size and time taken to sort, number of compares, number of swaps and number of hits using the above data for Quick Sort Dual Pivot algorithm**
* **Tabular & Graphical representation for Quick Sort Dual Pivot with Instrumentation as False**

|  |  |  |
| --- | --- | --- |
| **QuickSortDualPivot Results with Instrumentation as false** | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Runs** |
| 10000 | 21.3 | 20 |
| 20000 | 87.9 | 20 |
| 25000 | 101.8 | 20 |
| 30000 | 204.7 | 20 |
| 35000 | 287.25 | 20 |
| 40000 | 362.5 | 20 |
| 65000 | 757.55 | 20 |

* **Tabular & Graphical representation for Merge Sort with Instrumentation as True**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MergeSort Results with Instrumentation as true** | | | | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Compares** | **Number of Swaps** | **Number of Hits** | **Number of Runs** |
| 10000 | 4.05 | 1984262 | 9756 | 5223776 | 20 |
| 20000 | 4.35 | 4178160 | 19531 | 11327630 | 20 |
| 40000 | 5.6 | 9115637 | 39000 | 24414978 | 20 |
| 80000 | 9.15 | 19732185 | 78218 | 52350578 | 20 |
| 160000 | 17.65 | 41978815 | 156482 | 111741290 | 20 |
| 320000 | 46.1 | 88741030 | 312901 | 237561938 | 20 |

* **Below are the log-log graphs plotted with respect to array size and time taken to sort, number of compares, number of swaps and number of hits using the above data for Merge Sort algorithm**
* **Tabular & Graphical representation for Merge Sort with Instrumentation as False**

|  |  |  |
| --- | --- | --- |
| **Merge Sort Results with Instrumentation as false** | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Runs** |
| 10000 | 3.9 | 20 |
| 20000 | 4.55 | 20 |
| 40000 | 4.85 | 20 |
| 80000 | 10.95 | 20 |
| 160000 | 19.6 | 20 |
| 320000 | 38.1 | 20 |

* **Tabular & Graphical representation for Heap Sort with Instrumentation as True**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **HeapSort Results with Instrumentation as true** | | | | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Compares** | **Number of Swaps** | **Number of Hits** | **Number of Runs** |
| 10000 | 93.45 | 5614248 | 3017822 | 23299784 | 20 |
| 20000 | 452.7 | 12150727 | 6487934 | 50253190 | 20 |
| 40000 | 1770.6 | 26149307 | 13908549 | 107932810 | 20 |
| 80000 | 10315.9 | 56005239 | 29691232 | 230775406 | 20 |
| 160000 | 43836.9 | 119440830 | 63021182 | 490966388 | 20 |
| 320000 | 202829.05 | 253734017 | 133386636 | 1041014578 | 20 |

* **Below are the log-log graphs plotted with respect to array size and time taken to sort, number of compares, number of swaps and number of hits using the above data for Heap Sort algorithm**
* **Tabular & Graphical representation for Heap Sort with Instrumentation as False**

|  |  |  |
| --- | --- | --- |
| **Heap Sort Results with Instrumentation as false** | | |
| **Number of Elements (n)** | **Time taken in ns (t)** | **Number of Runs** |
| 10000 | 3 | 20 |
| 20000 | 4.4 | 20 |
| 40000 | 8.75 | 20 |
| 80000 | 15.35 | 20 |
| 160000 | 25.75 | 20 |
| 320000 | 61.15 | 20 |

From the above graphs plotted by considering arrays with different length it is evident that logarithmic graphs for time to sort and number of elements most closely matches with the logarithmic graphs for number of elements and number of comparisons. Therefore, comparisons are best predictor.