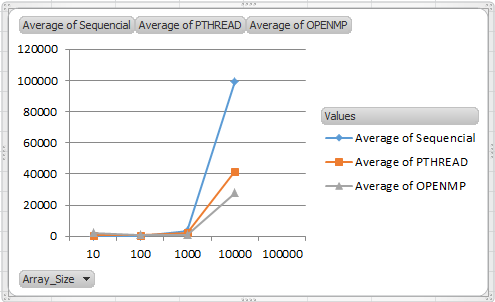
**Project Conclusion**

After executing the programs, the sorting times of each technique implemented on a sorting algorithm is compared.

The technique which gives the lowest time is considered as the best technique, and the one which gives the highest time is considered as the worst.

**Comparative Graphs**

**Binary\_Insertion Sort(Clock\_Time)**

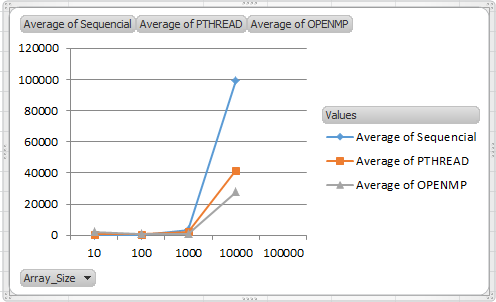


|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **Sequential** | **PTHREAD** | **OPENMP** |
| 10 | 3 | 709 | 2382 |
| 100 | 58 | 598 | 817 |
| 1000 | 3622 | 2288 | 1017 |
| 10000 | 99434 | 41588 | 27861 |
| 100000 | 9248780 | 2157261 | 2216668 |

**(Burst Time)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **Serial** | **PTHREAD** | **OPENMP** |
| 10 | 7 | 647 | 2425 |
| 100 | 63 | 459 | 960 |
| 1000 | 3583 | 598 | 1017 |
| 10000 | 99172 | 78097 | 27861 |
| 100000 | 9247398 | 4290962 | 2216668 |

**Selection Sort(Clock\_Time)**



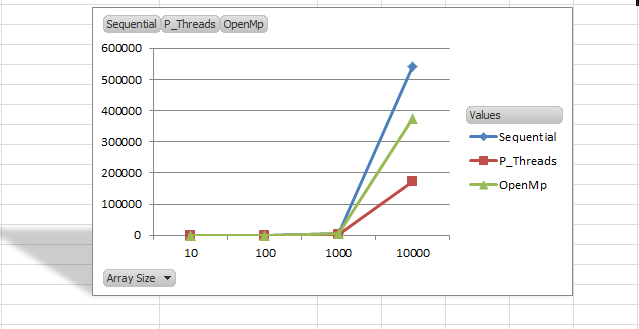
|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **Serial** | **Pthread** | **OpenMp** |
| 10 | 1 | 321 | 97 |
| 100 | 33 | 360 | 167 |
| 1000 | 3375 | 1836 | 2813 |
| 10000 | 233583 | 174578 | 182367 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **sequential** | **PTHREAD** | **OPENMP** |
| 10 | 1 | 148 | 12 |
| 100 | 35 | 340 | 59 |
| 1000 | 2606 | 14240 | 2436 |
| 10000 | 210689 | 631466 | 194323 |
|  |  |  |  |

**(Burst\_Time)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **SERIAL** | **PTHREAD** | **OPENMP** |
| 10 | 4 | 308 | 101 |
| 100 | 37 | 347 | 172 |
| 1000 | 5656 | 3815 | 2514 |
| 10000 | 160351 | 141579 | 94925 |

**Cocktail Sort(Clock\_Time)**



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential** | **P\_Threads** | **OpenMp** | **Array Size** |
| 1 | 450 | 125 | 10 |
| 32 | 731 | 269 | 100 |
| 4546 | 1929 | 39144 | 1000 |
| 541007 | 71912 | 419953 | 10000 |

**(Burst\_Time)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array\_Size** | **Serial** | **Pthread** | **OpenMp** |
| 10 | 3 | 752 | 209 |
| 100 | 834 | 699 | 619 |
| 1000 | 4553 | 3218 | 3964 |
| 10000 | 573569 | 318419 | 509846 |

**Results**

The above results show that the primary factor of efficiency is the **size of the data**. The smaller the

data the better sequential performs. On the other hand, the larger the data the better **PTHREAD** and

**OPENMP** performs. The second factor is the **CPU burst time**; the results show that depending on

the algorithms, sequential tends to consume more CPU Burst time than parallel programming.