**Abstract**

In this experiment, it shows the performance of adding two single dimension arrays versus two multiple dimension arrays. Also, the performance for finding the middle value of an array regardless order of elements.

**Experiment:**

* First Experient is performing the addition between two single dimension arrays with the size range from 100 – 250,000, and the times (in nanoseconds) are presented in the table below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size | 100 | 10,000 | 40,000 | 80,000 | 90,000 | 100,000 | 160,000 | 250,000 |
| Time | 695 | 48953 | 293957 | 480584 | 668793 | 697548 | 1168393 | 1627071 |

* Second Experiment is performing the addition between two two-dimension arrays with the size also range from 100 – 250,000, and the times (in nanoseconds) are presented below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size | 100 | 10,000 | 40,000 | 80,000 | 90,000 | 100,000 | 160,000 | 250,000 |
| Time | 818 | 64635 | 349308 | 321034 | 511934 | 627390 | 825348 | 1250464 |

After finishing these two experiments and comparing the numbers from the tables:

* The results show that the performance of adding two single dimension arrays for the size below 80,000 are faster than adding two 2D arrays.
* However, when the size reaches 80,000 the performance of adding two multidimension arrays are faster than adding two single dimension arrays.
* Third Experiment is finding the middle value of 1D array, regardless of order, basically the array would be cut into half and access the middle value. If the size of array is even then the middle value would be the average of midth and (mid + 1)th. Times for returning the middle value of 1D array in nanoseconds are presented in the table below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Size | 20 | 100 | 1000 | 10000 | 80,000 | 100,000 | 250,000 |
| Time | 93 | 104 | 100 | 99 | 109 | 98 | 104 |

The results show that the time for this performance is bounded between 100 nanoseconds with the boundaries of 10 nanoseconds for both upper bound and lower bound.

**Conclusion**

If the array sizes are smaller than 80,000, C++ handles single dimension arrays better than multiple dimension arrays but if the size is bigger than 80,00, C++ handles the add operation between two multidimension arrays better than single dimension arrays.