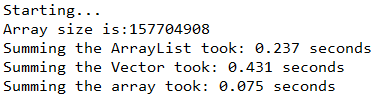
Efficiency tests for summing large collections

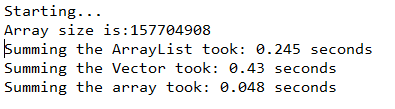
**Summary:**

In these experiments, a very large collection of randomly generated numbers will be summed from Vectors, ArrayLists, and Arrays to see which structure is the most efficient. The size of the large collection is determined by adding to the ArrayList in a for-loop for approximately 10 seconds.

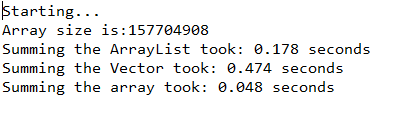
Trial 1:



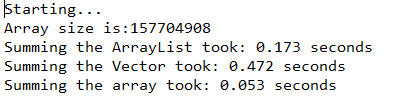
Trial 2:



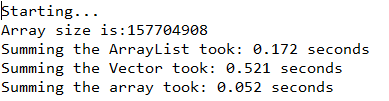
Trial 3:



Trial 4:



Trial 5:



**Average time amongst all trials:**

ArrayList: 0.201 seconds

Vector: 0.466 seconds

Array: 0.055 seconds

**Variance between max and min times:**

ArrayList: 0.073 seconds

Vector: 0.091 seconds

Array: 0.023 seconds

**Conclusion:**

For reasons unknown, ArrayList sum time began to shorten as the trials went on but stabilized around 0.170 seconds. The Vector summing showed the most variance without a discernable pattern as to why, from 0.521 seconds to 0.43 seconds, around a 20% difference. The fastest structure to sum 157704908 numbers is the array by an order of magnitude, however it also showed the largest variance between min and max times, 0.023 seconds, which is a 48% increase in the fastest time.

Possible explanations for the variance in the ArrayLists and Vectors could be the fact that iterators were used for ArrayList and Vectors, or ArrayLists and Vectors are dynamically sized. The Array did not use an iterator, and summed through a for loop, by index.

Despite the large variance, I would still recommend the Array to sum large numbers.