Anas Aly

CSCI 142

Dr. Basar

I implemented three different data structures in Java. Arrays, Arraylists and Vectors. I generated 50000000 **different** random numbers and added those numbers as elements to the three different data structures and sorted this data, each in a separate “for” loop. Then I kept track of the time it takes for each loop to be done. This way, I was able to compare the performance of each type of data storing interface/collection doing a similar task. I used OpenJDK 18 SDK. The code was run on IntelliJ IDEA 2022.2.1, A Windows 11 Home 22H2 machine with an Intel Core i5-11300H Processor.

Adding Elements

Array	**Array List**	**Vector**
765 milliseconds	2298 milliseconds	3158 milliseconds
0.765 seconds	2.298 seconds	3.158 seconds
100%	300.39%	412.81%

At first glance, about 2.3 seconds for an array list and 3.16 seconds for vectors does not seem like a lot of time. But by determining the time it takes an array to add elements to it as the baseline for comparison. We find that going from Arrays to Array Lists makes the time increase exponentially. An Array list for example takes 300 times the amount of time it takes for an array to do a similar task. This suggests that using an array for this task might be the ideal choice. If we have to choose between Array lists and Vectors, Array lists might be preferred rather than vectors for this task since it takes less time even if it is much slower than arrays. But in general, Arrays are fastest, Array lists are much slower than Arrays, and Vectors are somewhat slower than Array Lists.

Sorting Elements

Array	**Array List**	**Vector**
5715 milliseconds	35913 milliseconds	34966 milliseconds
5.715 seconds	35.913 seconds	34.966 seconds
100%	628.40%	611.83%

We notice a slightly different trend for the task of sorting elements. While arrays are also the fastest for doing this task. But vectors now are the second fastest, being slightly faster than Array lists and a bit slower than Arrays. We also notice that for the task of sorting, the difference grows exponentially between Arrays and other types of data collections discussed in this project. That an Array list is about 628 times slower than an Array for doing a similar task.

Console output

“Time to add elements to array: 765 milliseconds

Time to sort array: 5715 milliseconds

Time to add elements to array list: 2298 milliseconds

Time to sort array list: 35913 milliseconds

Time to add elements to vector: 3158 milliseconds

Time to sort vector: 34966 milliseconds

Process finished with exit code 0”

Conclusion

While arrays are usually faster for adding and sorting elements, we are limited by it being static in Java and so we have to define its dimension. In comparison, vectors and arraylists are dynamic and do not need to have a dimension defined. Thus, if there is a situation where we know exactly how many elements we are going to add in the future, arrays might be the ideal option for adding and sorting. Otherwise, Array lists might be ideal for the specific task of adding elements and vectors might be ideal for the specific task of sorting elements. The difference in sorting between Arrays and others is very noticeable, based on this data, we would only suggest using Array Lists or Vectors rather than Arrays in the case that we need a dynamic data collection. Vectors are preferred over Arraylists if there is a need for synchronization features of vectors.