	Array List	**Vector**	**Array**
Creation (10 Runs with 671,000,000 elements)	Run 1: **7.196960917s** Run 2: **6.936198375s** Run 3: **8.983641875s** Run 4: **6.769467583s** Run 5: **8.942093833s** Run 6: **7.645587125s** Run 7: **7.089619708s** Run 8: **7.309762083s** Run 9: **7.310750792s** Run 10: **7.286580417s**	Run 1: **8.637935167s** Run 2: **8.47299425s** Run 3: **6.868520625s** Run 4: **6.117909625s** Run 5: **6.854957583s** Run 6: **6.354540625s** Run 7: **6.12001975s** Run 8: **6.107750958s** Run 9: **6.36168025s** Run 10: **6.3495805s**	Run 1: **76.506127209s** Run 2: **77.880181458s** Run 3: **84.7087315s** Run 4: **84.962491958s** Run 5: **67.763356s** Run 6: **78.818063833s** Run 7: **80.157386s** Run 8: **80.14143075s** Run 9: **77.91921175s** Run 10: **80.635623417s**
Creation (Avg over 10 runs)	**7.5470662708s**	**6.824588933299999s**	**78.9492603875s**
Sum (250,000,000 elements)	Array List Sum: 1000027423 Sum Time: **0.133769041s**	Vector Sum: 999978207 Sum Time: **2.354658458s**	Array Sum: 1000038279 Sum Time: **0.085614209s**

From the table above, we can see that the longest creation time on average comes from the array, while the shortest creation time comes from a vector. When summing elements, the fastest time comes from an array, while the longest time comes from a vector.

For a recommendation to designers, it would seem that it depends on the task that needs to be performed. If the goal is to store many elements, then it would seem that a Vector works best. If the goal is to sum elements, it would seem that the use of an Array would be best. However, if the goal is to do both things and more, it would seem that the best overall recommendation is the use of an Array List. This is because the Array List has the best average of the 2 methods.