1. Assuming array size is 200, the sorting method that was fastest was SelectionSort (0.155 ms). MergeSort took 0.766 ms and BubbleSort took 0.381 ms.

2. Using the same array, isUnique2 was faster than isUnique1. isUnique2 took 0.003 ms and isUnique1 took 0.239 ms.

3. isUnique1 currently times out for arrays with size greater than 40 when it’s pre-sorted (and 300 when it’s not sorted). Using an array of size 30, isUnique2 is faster than isUnique1. isUnique2 took 0.005 ms while isUnique1 took 5019.14 ms.

4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | isUnique1 | isUnique2 | isUnique3  (BubbleSort) | isUnique3  (MergeSort) | isUnique3  (SelectionSort) |
| 100 | 0.007 | 0.003 | 0.258 | 0.376 | 0.11 |
| 500 | 7.235 | 0.005 | 6.335 | 1.889 | 2.417 |
| 1000 | N/A  (Timed Out) | 0.076 | 26.077 | 3.892 | 9.54 |
| 2000 | N/A  (Timed Out) | 0.057 | 104.554 | 7.941 | 37.875 |
| 3000 | N/A  (Timed Out) | 0.055 | 232.85 | 12.767 | 85.114 |
| 4000 | N/A  (Timed Out) | 0.391 | 421.936 | 16.212 | 150.943 |
| 5000 | N/A  (Timed Out) | 0.015 | 656.088 | 22.058 | 235.972 |

5. The time for isUnique1 varies greatly depending on the sequence and size of elements contained in the vector. On average, isUnique1 runs in under a minute with about 175 elements.

isUnique2 runs for just under a minute with a size of 900,000,000 elements.

isUnique3 with Bubble Sort runs in under a minute with a size of 85,000 elements.

isUnique3 with Merge Sort runs in under a minute with 14,000,000 elements.

isUnique3 with Selection Sort runs in under a minute with 160,000 elements.