Analysis of Lookup Time for Hashtable vs. Lookup Time for Array

Procedure:
The process I followed for analyzing lookup time was to load a file of one million, a file of three million, and a file of five million strings onto the heap. The pointers to these strings were loaded into both an array and into my hash table. Then using a random selection of one hundred strings from the larger list, I searched for the one hundred items and timed the computer in microseconds.

Each lookup was run five times. The averaged data was plotted to show the average lookup time for each size list.

Data:

Time 1m (us)

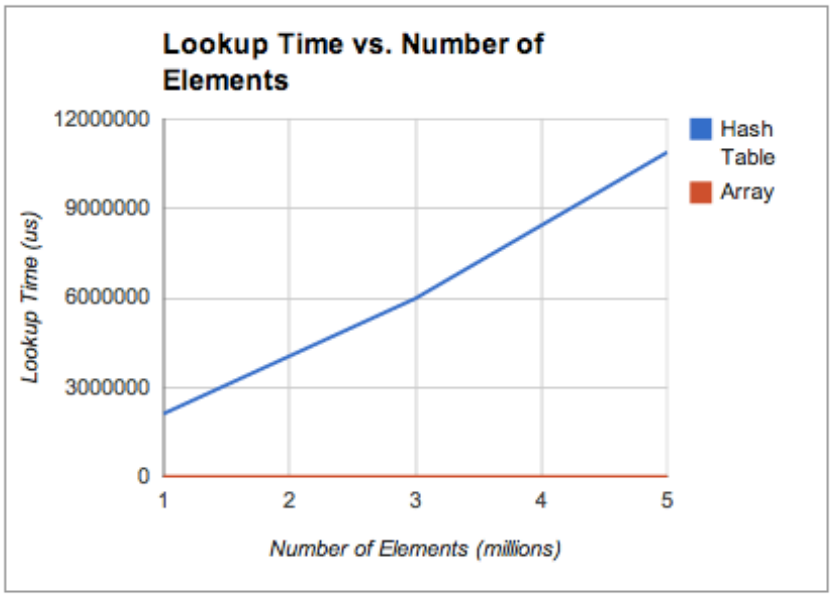
	Array	Hash Table
Run1	2190810	93
Run2	2090050	84
Run3	2194550	80
Run4	2055370	90
Run5	2046070	79
Ave	2115370	85.2

Time 3m (us)

	Array	Hash Table
Run1	6104060	80
Run2	5939850	83
Run3	5971950	82
Run4	5940730	82
Run5	5953210	81
Ave	5981960	81.6

Time 5m (us)

	Array	Hash Table
Run1	10828000	94
Run2	10817100	80
Run3	10949600	84
Run4	10858900	94
Run5	10975500	77
Ave	10885820	85.8



Conclusion:
Looking at the chart above, it is clear that lookup time for a hash table is O(1). It is also clear that the lookup time for an array increases linearly O(N) with the number of elements. This shows that hash tables are much better at looking up data than an array. Hash tables are a great way to index large volumes of data.