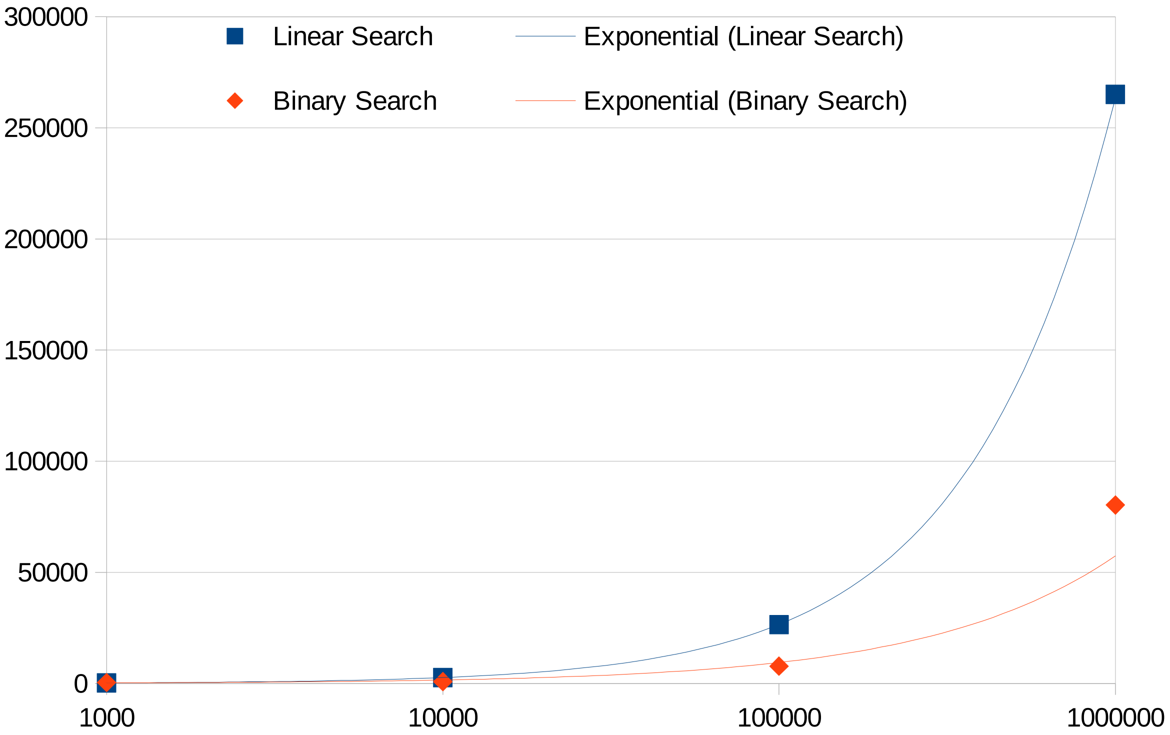
n	Time (Linear Search)	Time (Binary Search)
1000	273	415
10000	2653	831
100000	26480	7804
1000000	265068	80323

Linear search vs binary search run times. All times reported are in microseconds.



Linear search is faster than binary search when small data sets are used, as it takes some time to construct a binary search tree. However, the overhead involved is quickly dwarfed as the size of the array increases, with binary search becoming over four times as fast when one-million elements are used. This is because linear search grows linearly, as each element in the array has to be checked every time. Binary search, on the other hand, grows logarithmically, at a rate of O(logn), because it only has to check some branches of the tree to find the search item. However, binary search takes a lot more effort to develop, requiring the infrastructure of binary search trees to function, whereas linear search can be coded with your eyes closed. Linear search also uses less memory, if the haystack isn't in a binary search tree to begin with. Because of those reasons, linear search might be worth using on small (sub 2000 item) arrays, but binary search is preferred for large data sets.