



Algorithm Descriptions:

Linear Search:

The simplest of these three searches, a linear search simply traverse an array from the start to the end (or until it finds what it is looking for). The time complexity for this type of search algorithm is $O(n)$ since the complexity is directly related to the size of n .

Binary Search:

This is a much faster algorithm than linear search while still being relatively simple. It has a time complexity of $O(\log n)$ because it basically splits in two the array (or whatever is being searched) with every iteration. This is because it compares the middle value (or any value really) and then sorts based on values higher and lower than that one.

Binary Recursive Search:

Binary Recursive Search is very similar to the above binary search, just that uses recursion instead of iteration to run. This has the same time complexity as the binary search algorithm, but the space complexity is worse. The space complexity is $O(\log n)$ while Binary Search (iterative) has space complexity of $O(1)$.

