# Linear Search:

Theory:

Requires a dataset, list (could be numbers, words, ...). Doesn't need to be ordered.

Requires a target to search for within the list, we will use X.

Starts from left to right (low to high index), compare the element at each index with X.

If list[n] == X return the index where X is at.

Else go to the next index (n+=1) and do the same comparison, repeat until found.

Otherwise return -1 (null) which indicates the value isn't in the list.

Big O

Time Complexity

Worst case: O(n)

Average case: O(n)

Best case: O(1)

# Binary Search:

Works on an ordered list.

Requires a target to search for.

Divide and conquer, discard half of the list each time.

When the middle term is the target or you’re left with the target, job done.

Big O

Time complexity

Worst case: O()

Average case: O(log n)

Best case: O()