**Linear vs. Binary Search**

( **Linear Search** )

Techniques used to search arrays include the linear search, also known as a sequential search. For the linear search approach, a search is performed through an element by element comparison basis, from either the beginning to the end or from the end to the beginning. For this type of model the list may be in any order, i.e. the list does not have to be in a sorted order.

**(Binary Search)**

Steps

To analyze the binary search algorithm, leaf thru each comparison of a sorted array to eliminate about half of the remaining items from consideration till you find your target value. If you start with *n* items, *about* n/2 items will be left after your first comparison. After your second comparison, there will be *about* n/4 items left and so on till your left with one item to find if your target is found or not.

-Sample algorithm to find a given targeted value if it exists in a list

**boolean bSearch(A, target)**

lo = 0, hi = n - 1

while lo <= hi

mid = (hi+lo)/2

if A[mid] == target:

**return** **true**

else if A[mid] < target:

lo = mid+1

else:

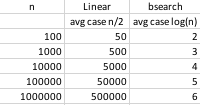
hi = mid-1

**return false** // target was not found

BS Time complexity

Best case O(1) / Avg. Case O(log n) / Worst Case O(log n)

Speed comparisons



BSearch process example

What would be the maximum number of comparisons needed for an array list with elements consisting of **{1,2,5,10,15,21}** items and a target value of **2**? Circle the appropriate amount of comparisons shown below as your answer.

|  |  |
| --- | --- |
| **Comparisons** | **n items left to compare in list** |
| 1 | n/2 |
| 2 | n/4 |
| 3 | n/8 |
| 4 | n/16 |