

BINARY SEARCH



Binary Search

- Binary Search
- Implementation of Binary Search
 - Iterative
 - Recursive

Binary Search

- We can take greater advantage of the ordered list!
- Instead of searching the list in sequence, a **binary search** will start by examining the middle item.

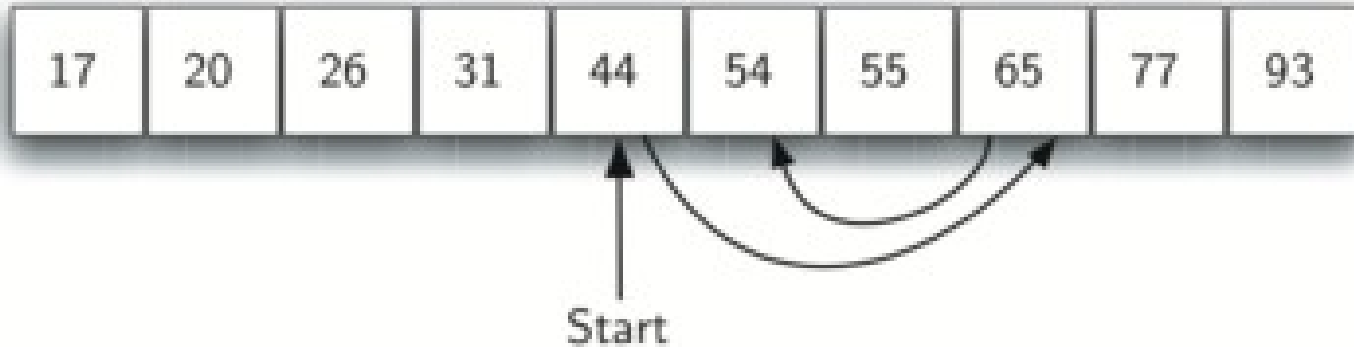
- A **binary search** will start by examining the middle item.
- If that item is the one we are searching for, we are done.
- If the item we are searching for is greater than the middle item, we know that the entire lower half of the list as well as the middle item can be eliminated from further consideration.
- The item, if it is in the list, must be in the upper half.

Binary Search

- We can then repeat the process with the upper half. Start at the middle item and compare it against what we are looking for.
- Again, we either find it or split the list in half, therefore eliminating another large part of our possible search space.

Binary Search

- For example, searching for **54** again:



Divide and Conquer

- Binary search uses Divide and Conquer!
- We divide the problem into smaller pieces, solve the smaller pieces in some way, and then reassemble the whole problem to get the result.

Binary Search Analysis

- Each comparison eliminates about half of the remaining items from consideration.
- What is the maximum number of comparisons this algorithm will require to check the entire list?

Binary Search Analysis

Comparisons	Approximate Number of Items Left
1	$\frac{n}{2}$
2	$\frac{n}{4}$
3	$\frac{n}{8}$
...	
i	$\frac{n}{2^i}$

Binary Search Analysis

- Up next, implementing two versions of Binary Search!