Binary Search Problem (D&C Approach)

- The algorithm is used to look up a word in a dictionary or a name in a telephone directory.
- Let T [1…n] be an array sorted in ascending order.
- Let x be some item. The problem consists of finding x in the array. If x is not in the array, then find the position where it might be inserted.
- **1st Approach: Sequentially**
  - **Function sequential (T [1..n], x)**

    {Sequential search for x in the array T}

    for i <- 1 to n do

    if T[i] >= x then return i

    return n +1
- Worst case Ώ(n) and Best case O(1), time is Θ (n)

Binary Search Problem (2nd Approach)

- **Use D & C Approach:**
  - To Speed Up the search, look for x either in first half or in the second half of an array.
  - Let k=[n/2] ; If x <= T[k], then search x in T [1…k] ; other wise search in T[k+1….n]

Binary Search Problem (2nd Approach)

- **Algorithm:**

Function binsearch (T[1…n], x)

if n=0 or x>T[n] then return n+1

else return binrec(T[1…n], x)

Function binrec(T[i…j], x)

{ Binary search for x in subarray T[i…j] with promise that T[i-1] < x <= T[j] }

if i=j then return i

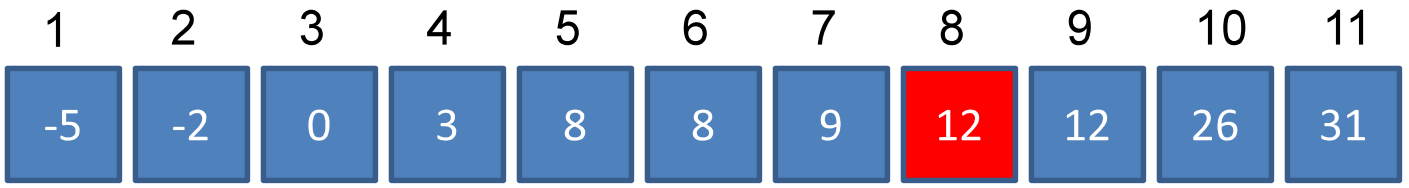
k <- (i+j)/2

if x <= T[k] then return binrec(T[i…k], x)

else return binrec(T[k+1….j], x)

Binary Search Problem (2nd Approach)

X=12



i k j		X <= T[k]
	i k j	No
i k j		Yes
ik j		Yes
ij		No
		i=j ; stop

Worst & Average Case: O (log N), Best Case: O(1)