

# LINEAR SEARCH

## 1. LOOP INVARIANT:-

Pseudocode:

Linear Search ( $A, v$ )

for  $j = 1$  till  $A.length$ :

if  $A[j] == v$ :

return  $j$

return NULL

- The loop continues as long as we haven't found ' $v$ ' in the array i.e. the loop checks if  $A[j] == v$  at every iteration.
- At index  $j$  we haven't yet found the value  $v$ . So, the subarray  $A[1:j-1]$  does not contain the value  $v$ .

## 2. INITIALIZATION:

It is true prior to the first iteration of for loop.

- When  $j = 1$  i.e. at the very first element, before that the array  $A[1:j-1]$  is **empty**. It does not contain value  $v$ .

## 3. MAINTENANCE:

If it is true before an iteration of the loop, it remains true before the next iteration. The loop keeps on checking if  $A[1:j-1]$  until it finds  $v$ , hence it maintains the for loop.

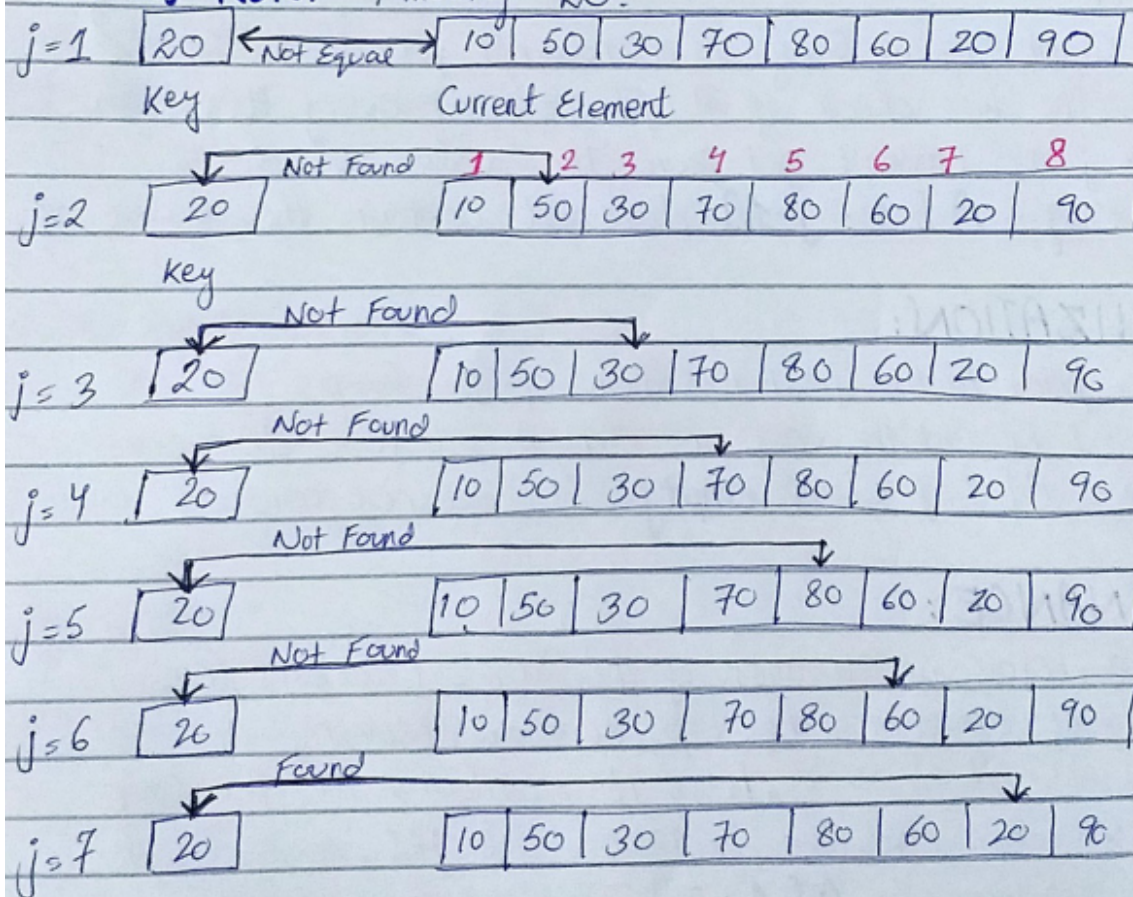
if  $A[j] == v$ ; this condition continues till  $j+1$ , moving towards right at each iteration.  $A[1:j]$  does not contain value  $v$ .



#### 4. TERMINATION:-

The for loop terminates when  $j > A.length$ . The length of Array A is  $n$  and during termination  $j = n+1$ . By substituting this in  $A[1:j-1] \Rightarrow A[1:n]$  indicates that the original value is not present in Arrays and return NULL.

**DRY RUN:-** Finding 20.



Since the key is found, return its index i.e.  $j=7$



# INSERTION SORT

## 1. LOOP INVARIANT:-

Pseudocode:

for  $j=2$  to  $A.length$

$key = A[j]$

$i = j-1$

    while  $i > 0$  and  $A[i] > key$

$A[i+1] = A[i]$

$i = i-1$

$A[i+1] = key$

"At the start of each iteration of for loop, the subArray  $A[1 \dots j-1]$  consists of elements originally in  $A[1 \dots j-1]$ , but in sorted order."

## 2. INITIALIZATION:

When  $j=2$ , then the subArray  $A[1 \dots j-1]$  only consists of a single element  $A[1]$ ; which is the original element in  $A[1]$  and it is in sorted order.

## 3. MAINTENANCE:

Each iteration maintains loop invariant. The for loop moves  $A[j-1]$ ,  $A[j-2]$ ,  $A[j-3]$  to the right. Hence, by substituting  $j+1$  in the subArray  $A[1 \dots j-1]$ , we get  $A[1 \dots j]$  consists of elements in sorted order.



#### 4. TERMINATION:

The condition that causes the for loop to terminate  $j > a.length = n$ .  
 We get  $j = n+1$  at termination. Substituting  $n+1 = j$  in subArray  $A[1 \dots j-1]$  we get  $A[1 \dots n]$  is the entire Array that consists of elements in sorted (ascending) order.

#### DRY RUN

	-1	0	1	2	3	4	5	→ indexes
Key = 54, j = 1, i = 0	12	54	65	7	23	9		
		compare						
Key = 65, j = 2, i = 1, 0	12	54	65	7	23	9		
Key = 7, j = 3, i = 2, 1, 0	12	54	65	7	23	9		
			compare and swap					
" i = 1 "	12	54	65	65	23	9		key being compared
" i = 0 "	12	54	54	65	23	9		
" i = -1 "	12	12	54	65	23	9		
" i = -1 "	7	12	54	65	23	9		
Key = 23, j = 4, i = 3, 2, 1, 0	7	12	54	65	23	9		
i = 3	7	12	54	65	65	9		
i = 2	7	12	54	54	65	9		
i = 1	7	12	23	54	65	9		
	7	12	23	54	65	9		
Key = 9, j = 5, i = 4, 3, 2, 1	9	7	12	23	54	65		