

Arr[9]

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
33	55	88	77	44	11	66	22	99

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

- 1) Get the element(key) to search from the user.
- 2) Start the traversal from the 1st element (0th index)
- 3) Compare the key with each element in the array
 - a. If key is found, return the index
 - b. Else, continue the search till the last index

Key = 77
Key found at index 3
Comparisons = 4
Average case time complexity

$n/2 \rightarrow$ order of n
 $1/2$ is constant and in asymptotic analysis constant is discarded
Big theta $\Theta(n)$

Key = 33
Key is found at index 0
Comparison = 1
Best case time complexity

Order of $(1) \rightarrow o(1) \rightarrow$ *Big omega $\Omega(1)$*

Key = 99
Key is found at index 8
Comparisons = 9
Worst case time complexity

Order of $(n) \rightarrow$
 $o(n) \rightarrow n$ is the number of elements
Big oh $\rightarrow O(n)$

Key = 110
Key not found
Comparisons = 9
Worst case time complexity

Order of (n)
 $o(n)$
 N is the number of elements
Big oh $\rightarrow O(n)$