



Sequential search:

- Here, the algorithm compares successive elements of a given list with a given search key until either a match is encountered (successful search) or the list is exhausted without finding a match (unsuccessful search). Here is an algorithm with enhanced version: where, we append the search key to the end of the list, the search for the key have to be successful, so eliminate a check for the list's end on each iteration.



Algorithm sequential search($A[0..n]$, K)

// Input: An array A of n elements & search key K .

// Output: The position of the first element in $A[0..n-1]$ whose value is equal to K or -1 if

// no such element is found.

$A[n] \leftarrow k$

$i \leftarrow 0$

While $A[i] \neq K$ do

$i \leftarrow i+1$

if $i < n$ return i

else return -1 .



- Another , method is to search in a sorted list. So that the searching can be stopped as soon as the element greater than or equal to the search key is encountered.



Analysis:

- The efficiency is determined based on the key comparison

$$(1) C_{\text{worst}}(n) = n.$$

- when the algorithm runs the longest among all possible inputs i.e., when the search element is the last element in the list.

$$(2) C_{\text{best}}(n) = 1$$



Thank you!