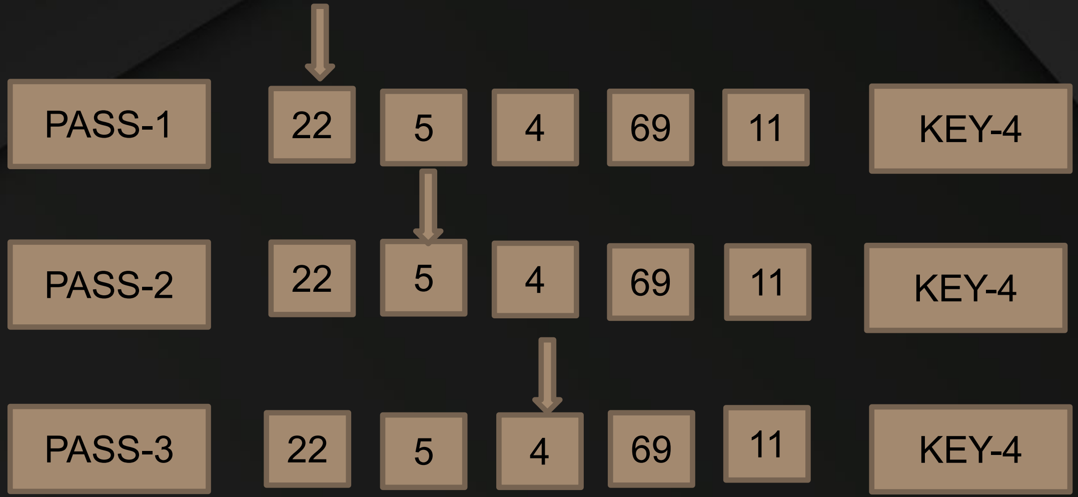
Searching & Sorting

Abhishek Makwana

Searching In array

1-Linear Search



PASS-1 22 5 4 69 11 KEY-4

PASS-2 22 5 4 69 11 KEY-4

PASS-3 22 5 4 69 11 KEY-4

Time Complexity

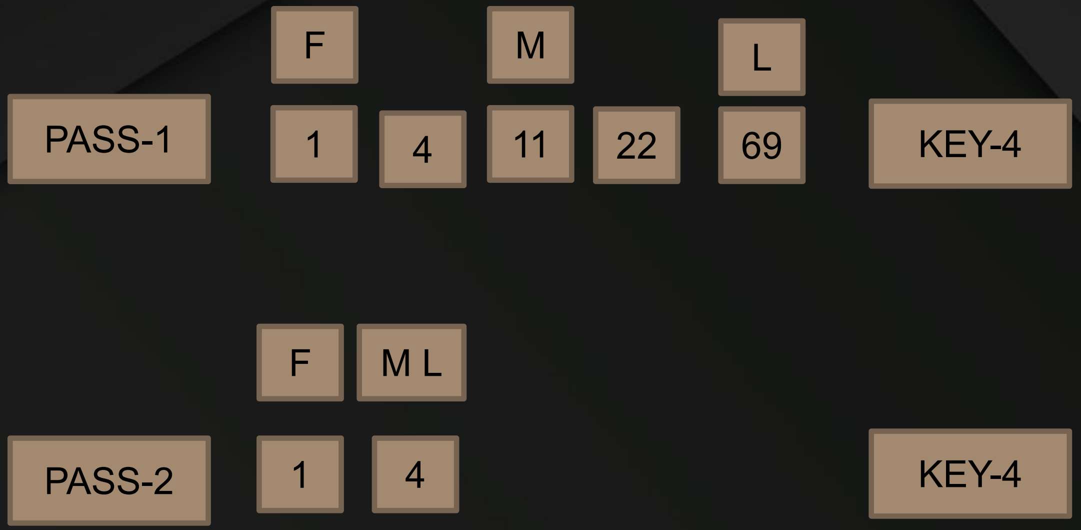
BEST
CASE
O(1)

AVERAGE
CASE
O(N)

Worst Case

O(N)

2-Binary Search



F M L

PASS-1 1 4 11 22 69 KEY-4

F M L

PASS-2 1 4 KEY-4

Time Complexity

BEST CASE

O(1)

AVERAGE CASE

O(Log N)

Worst Case

O(Log N)

3-Ternary Search

Array is divided into Three parts to reduce the time complexity

Time - Complexity

Best Case Complexity - O(1)

Average Case Complexity – O(Log n base 3)

Worst Case Complexity – O(Log n base 3)

Time Complexity

BEST CASE
O(1)

AVERAGE CASE
O(Log N Base 3)

Worst Case

O(Log N Base 3)

Sorting An Array

1-Bubble Sort

Each element is compared to its left/right most element

Time - Complexity

Best Case Complexity - O(N)

Average Case Complexity – O(N^2)

Worst Case Complexity – O(N^2)

Time Complexity

BEST CASE

O(N)

AVERAGE CASE

O(N^2)

Worst Case

O(N^2)

Details

STABLE
YES

IN
PLACE
Y ES

2-Selection Sort

Highest/Lowest element is selected and placed at starting of array and so on the array is sorted

Time - Complexity

Best Case Complexity - O(N^2)

Average Case Complexity – O(N^2)

Worst Case Complexity – O(N^2)

Time Complexity

BEST
CASE
O(N^2)

AVERAGE
CASE
O(N^2)

Worst Case

O(N^2)

Details

STABLE

NO

IN PLACE

YES

THANK YOU