**LEC 1 - SEARCHING TECHNIQUES**

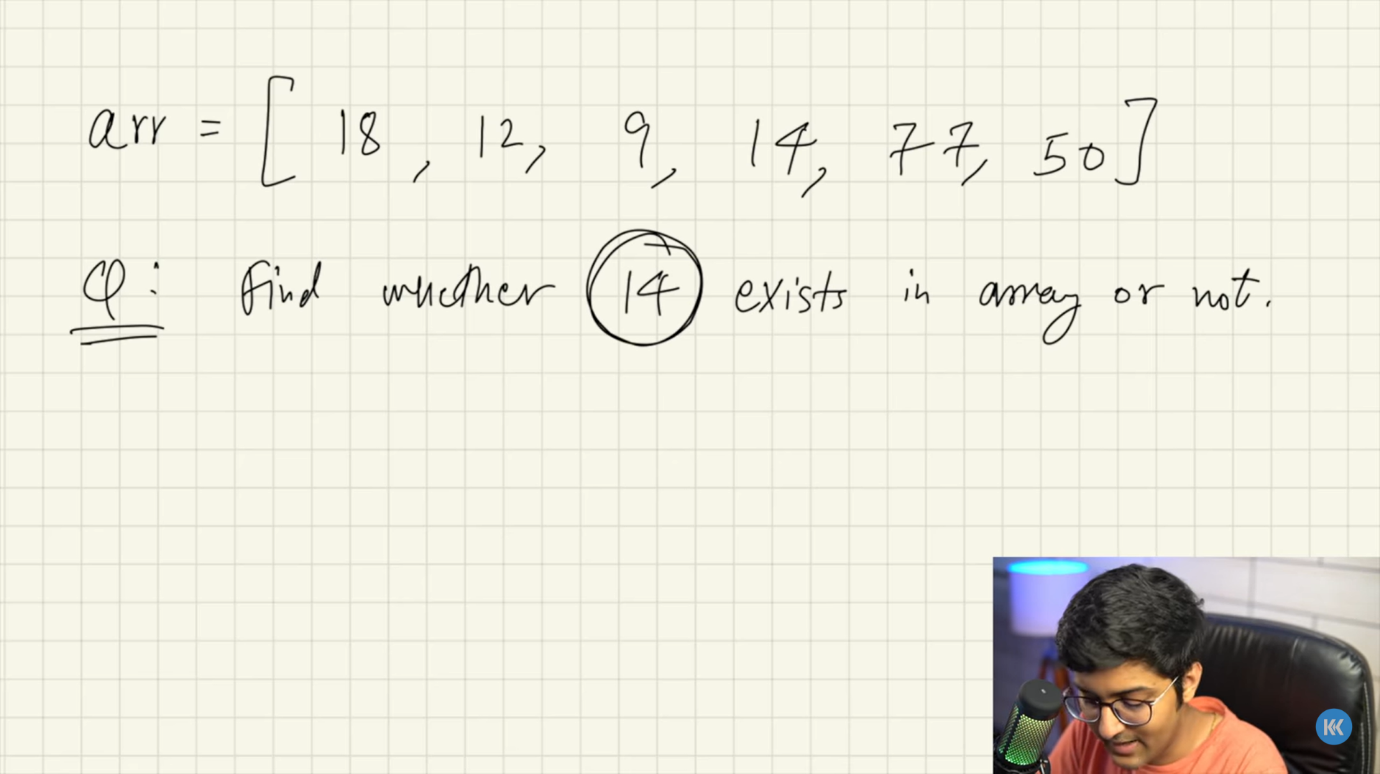
01-LINEAR SEARCH

What is a searching?

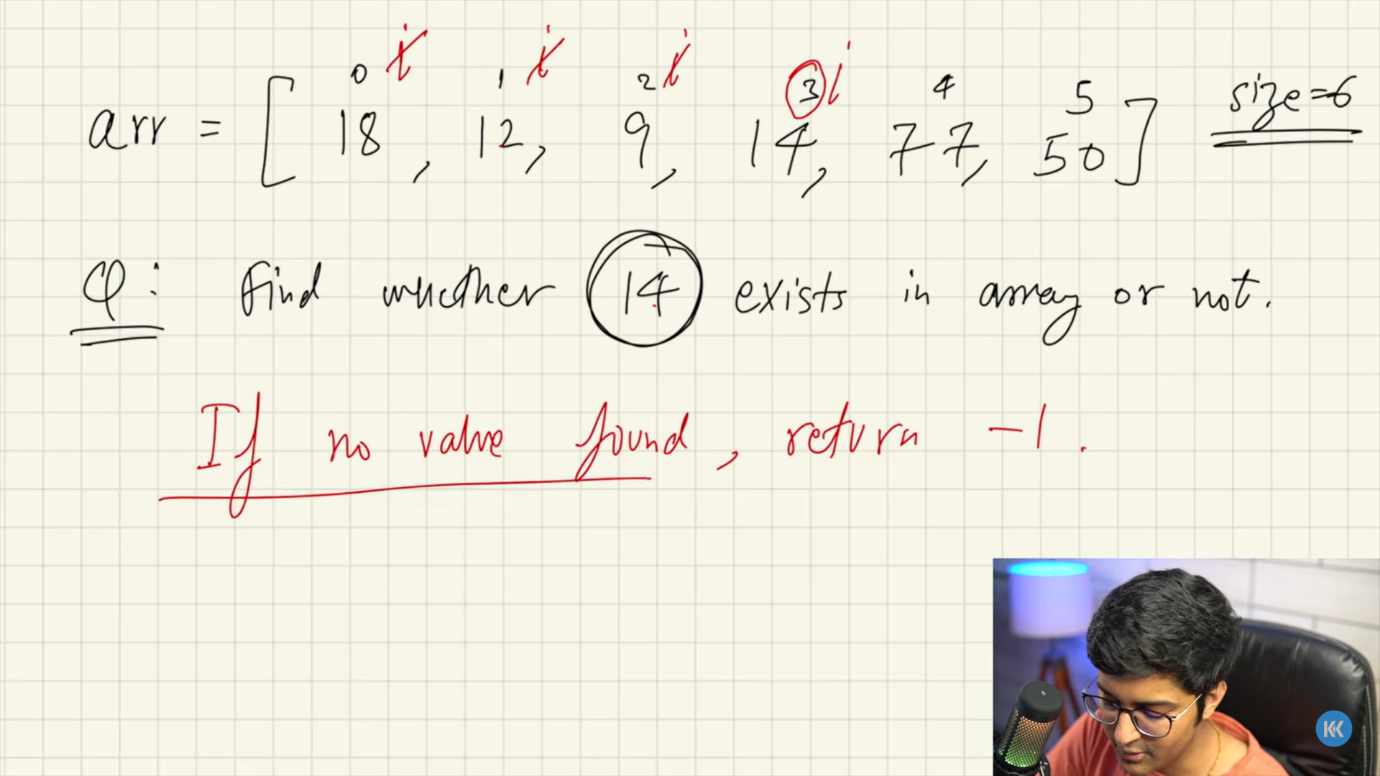
It is not only about searching . with this searching techniques we can modify remove replace or insert an element in an data .

This is one of the very most important concept in the DSA

BASIC QUESTION FOR AN SEARCH :



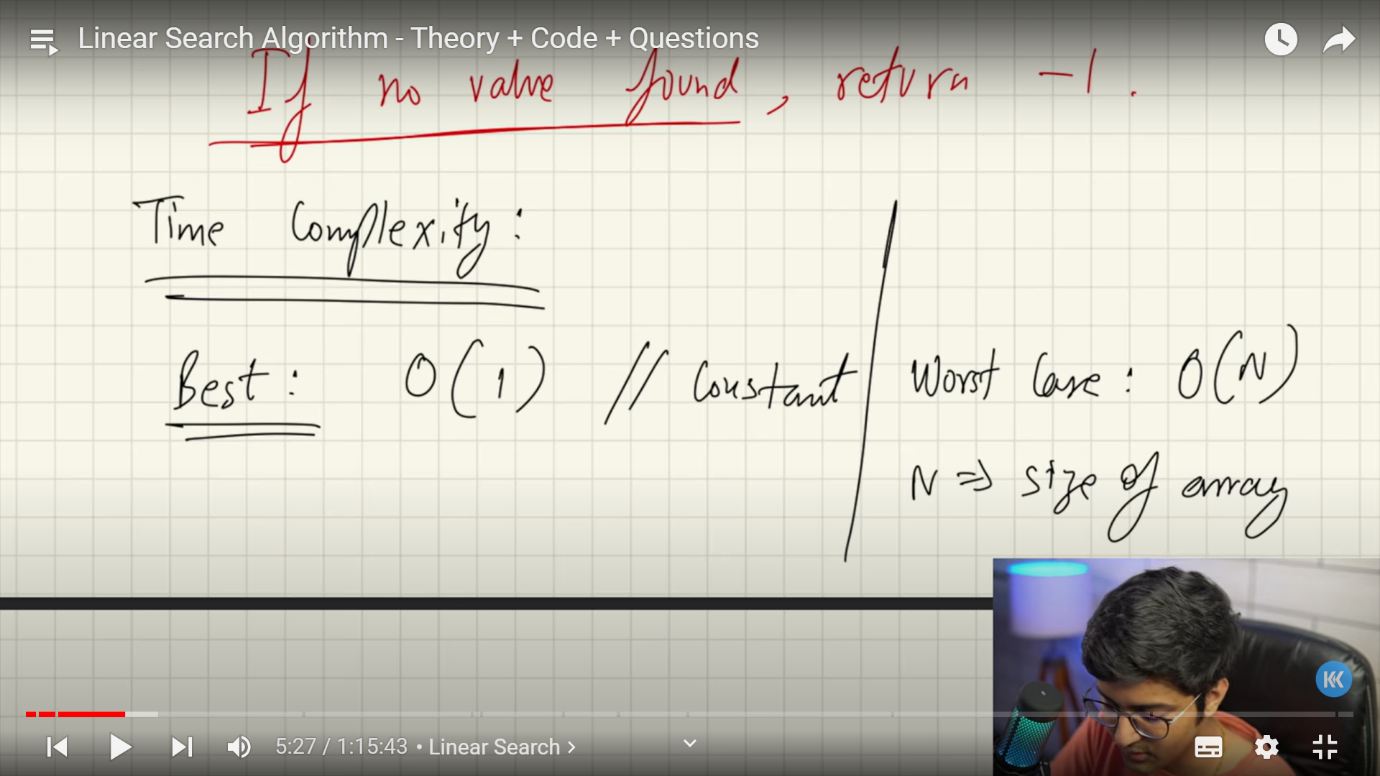
WHAT LINEAR SEARCH DOES:



Considering the first element as I and it will ask the condition , if equal it will return the index if not I will movw to the next value

.

BASIC OF TIME COMPLEXITY;

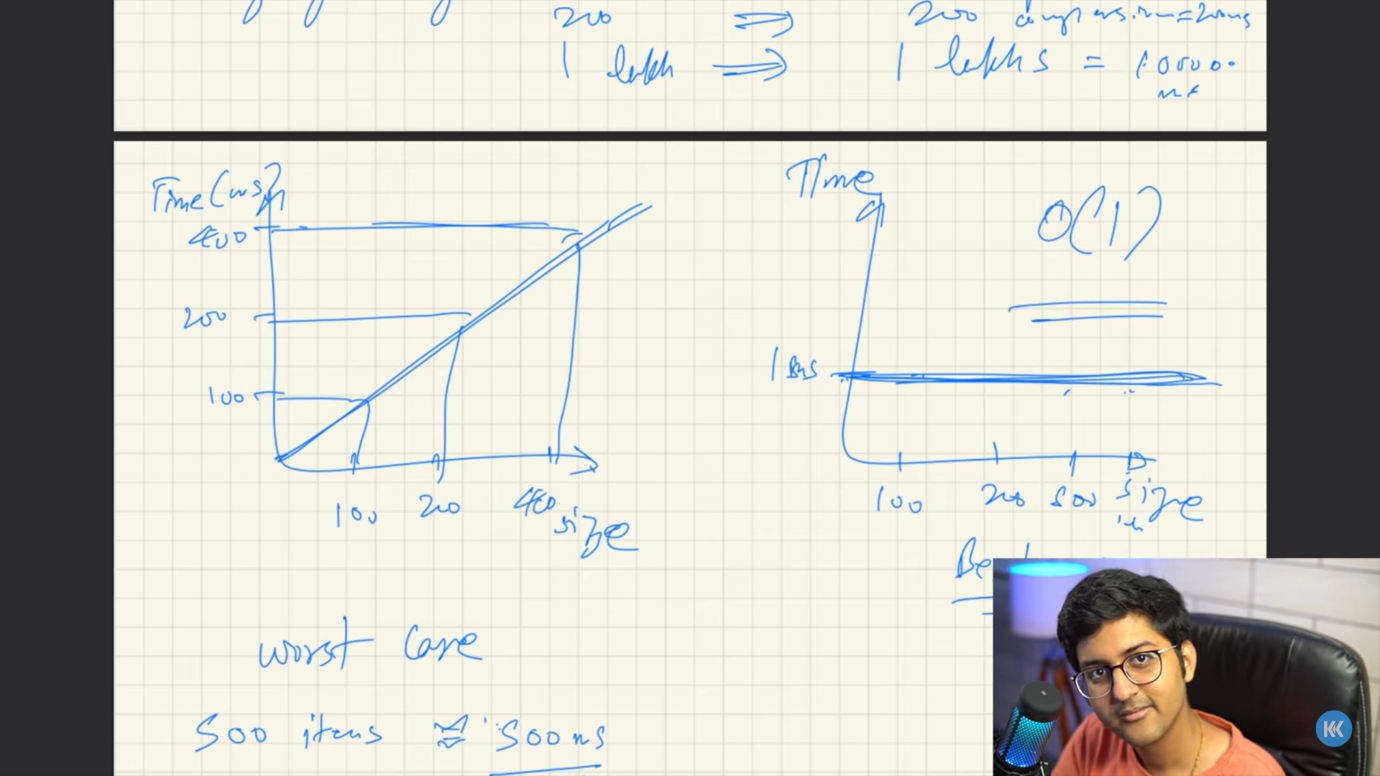


In normal words …

Best case is nothing but how many loop does it take to find that element. Not just the loop

Lets say we have an array of 200 elements the element we are searching for is in the 0th index this is best case it just takes 1 loop

Lets say the element you want to find is not in the whole 200 elements this will be said as the worst case .going through all the comparision loop and getting no result .



Worst case has a linear time complexity because when the size increases the time for the compilation also increases

.On the other hand in best case the time complexity is constant no matter what the size of the array is even the size is near thousand the time taken will be the same as the 200 element when the element is found in the 0th index.

FIRST PROGRAM ON LINEAR SEARCH:

To search an element in an integer type array:

package SEARCHING01\_LINEAR;  
  
import java.util.Scanner;  
  
public class LINEAR01\_INTEGER {  
 public static void main(String[] args) {  
 while (true) {  
  
 //FOR INTEGERS  
  
 Scanner value = new Scanner(System.*in*);  
 System.*out*.println("Enter the value to be searched in the array [ Exit enter 101 ] : ");  
 int search = value.nextInt();  
 if (search == 101) break;  
 int[] arr = {1, 3, 5, 7, 9, -5};  
 if (*search*(arr, search) < 0) System.*out*.println("no value " + *search*(arr, search));  
 else System.*out*.println("The value that you searched for is in the index " + *search*(arr, search));  
 }  
 }  
 static int search(int[] array, int tosearch) {  
 if (array.length == 0) {  
 return -1;  
 }  
 for (int i = 0; i < array.length; i++) {  
 if (array[i]== tosearch){  
 return i;  
 }  
 }  
 return -1;  
 }  
}

(NOTE : whenever a return is executed the function will be terminated and move to the main)

To search an element in string type:

package SEARCHING01\_LINEAR;  
  
import java.util.Scanner;  
  
public class LINEAR02\_STRING {  
 public static void main(String[] args) {  
 while (true){  
  
 //FOR STRING  
  
 Scanner value = new Scanner(System.*in*);  
 System.*out*.println("Enter the element to be searched in the array [ Exit enter Esc ] : ");  
 String search = value.next();  
 if (search.equals("Esc")) break;  
 String str = "yogesh";  
 System.*out*.println(*st\_search*(str , search));  
 }  
 }  
 static String st\_search(String var , String tosearch){  
 if (var.isEmpty()) System.*out*.println("is empty");  
 for (int i = 0; i < var.length(); i++) {  
 if (tosearch.charAt(0) == var.charAt(i)){  
 return ("the element entered index is "+ i );  
 }  
 }  
 return ("YOU ENTERED WRONG ELEMENT");  
 }  
}

To search an element of string and integer by giving range:

package SEARCHING01\_LINEAR;  
  
import java.util.Scanner;  
  
public class LINEAR03\_RANGE {  
 public static void main(String[] args) {  
 while (true){  
  
 //FOR STRING  
  
 Scanner value = new Scanner(System.*in*);  
 Scanner start = new Scanner(System.*in*);  
 System.*out*.println("Enter the starting and ending range with space : ");  
 int srt = start.nextInt();  
 int end = start.nextInt();  
 System.*out*.println("Enter the element to be searched in the array [ Exit enter Esc ] : ");  
 String search = value.next();  
 if (search.equals("Esc")) break;  
 String str = "yogesh";  
 System.*out*.println(*st\_search*(str , search , srt ,end));  
 }  
  
 while (true) {  
  
 //FOR INTEGERS  
  
 Scanner start = new Scanner(System.*in*);  
 System.*out*.println("Enter the starting and ending range with space : ");  
 int srt = start.nextInt();  
 int end = start.nextInt();  
 Scanner value = new Scanner(System.*in*);  
 System.*out*.println("Enter the value to be searched in the array [ Exit enter 101 ] : ");  
 int search = value.nextInt();  
 if (search == 101) break;  
 int[] arr = {1, 3, 5, 7, 9, -5};  
 if (*search*(arr, search ,srt ,end) < 0) System.*out*.println("no value " + *search*(arr, search, srt, end));  
 else System.*out*.println("The value that you searched for is in the index " + *search*(arr, search, srt, end));  
 }  
 }  
 static String st\_search(String var , String tosearch , int start , int end){  
 if (var.isEmpty()) System.*out*.println("is empty");  
 for (int i = start; i < end+1 ; i++) {  
 if (tosearch.charAt(0) == var.charAt(i)){  
 return ("the element entered index is "+ i );  
 }  
 }  
 return ("YOU ENTERED WRONG ELEMENT");  
 }  
  
 static int search(int[] array, int tosearch, int start , int end) {  
 if (array.length == 0) {  
 return -1;  
 }  
 for (int i = start; i < end; i++) {  
 if (array[i]== tosearch){  
 return i;  
 }  
 }  
 return -1;  
 }  
  
}

2D ARRAY SEARCH:

package SEARCHING01\_LINEAR;  
  
import java.util.Scanner;  
  
public class LINEAR05\_2DARRAY {  
 public static void main(String[] args) {  
 int[][] arr = {  
 {1 , 3 , 6 , 40 , 91},  
 {2 , 5 , 7 , 8 , 40},  
 {11 , 43 , 77 , 34 , 40}  
 };  
 int target = 40;  
 *search*(arr , target);  
  
 System.*out*.println(*maximum*(arr));  
  
 Scanner r1 = new Scanner(System.*in*); //array row size  
 System.*out*.println("Enter the no of row : ");  
 int[][] array = new int[r1.nextInt()][];  
 *element\_entry*(array);  
 }  
  
static void element\_entry(int[][] array){  
 for (int Rows = 0; Rows < array.length; Rows++) {  
 Scanner c1 = new Scanner(System.*in*);  
 System.*out*.println("Enter the coloum size for " + Rows + " row ");  
 int sw = c1.nextInt();  
 array[Rows] = new int[sw]; //array of this row size  
 for (int Colz = 0; Colz < sw; Colz++) {  
 Scanner array\_in = new Scanner(System.*in*);  
 System.*out*.println("Enter the " + Colz + " elements : ");  
 array[Rows][Colz] = array\_in.nextInt();  
 }  
 }  
 Scanner ne = new Scanner(System.*in*);  
 System.*out*.println("Enter the target : ");  
 *search*(array , ne.nextInt());  
}  
  
 static void search (int[][] array , int target){  
 int answer = 0;  
 int answer01 = 0 ;  
 for (int row = 0; row < array.length; row++) {  
 for (int col = 0; col < array[row].length; col++) {  
 if (array[row][col]==target){  
 answer= col;  
 answer01 = row;  
 }  
 }  
 }  
 System.*out*.println("The value you entered is in the "+answer01+" row and in column "+answer);  
 }  
  
 static int maximum(int[][] array){  
 int answer = array[0][0];  
 for (int[] ints : array) {  
 for (int col = 1; col < ints.length; col++) {  
 if (answer < ints[col]) {  
 answer = ints[col];  
 }  
 }  
 }  
 return answer;  
 }  
}

In 2D search there is 2 ways on a array which is already defined and the one we define ..both the methods are coded ,

EVEN NUMBERS SEARCH AS NUMBER OF ELEMENTS :

package SEARCHING01\_LINEAR;  
  
public class LINEAR06\_EVEN\_ODD {  
 public static void main(String[] args) {  
 int[] arr = {123561, 32, 5123, 735, 4576 , 96};  
  
 //Second method  
  
 System.*out*.println(*nam*(arr));  
  
 }  
 static int nam(int[] num){  
 int count = 0 ;  
 for (int from = 0; from < num.length; from++) {  
 String h = String.*valueOf*(num[from]);  
 if (*eve*(h)) {  
 count++;  
 }  
 }  
 return count;  
 }  
  
 static boolean eve(String k){  
 int h = k.length();  
 if (h % 2 == 0){  
 return true;  
 }  
 return false;  
 }  
}

2D- addition and comparision:

package SEARCHING01\_LINEAR;  
  
public class LINEAR07\_2D\_ADD {  
 public static void main(String[] args) {  
 int[][] arr = {  
 {1 , 2 , 5},  
 {4 , 6},  
 {3 , 1 , 1}  
 } ;  
 Solution call = new Solution();  
 System.*out*.println(call.maximumWealth(arr));  
 }  
}  
class Solution {  
 public int maximumWealth(int[][] accounts) {  
 int result2 = 0 ;  
 for (int[] account : accounts) {  
 int result1 = 0;  
 for (int col : account) {  
 result1 += col;  
 if (result1 > result2) {  
 result2 = result1;  
 }  
 }  
 }  
 return result2;  
 }  
}