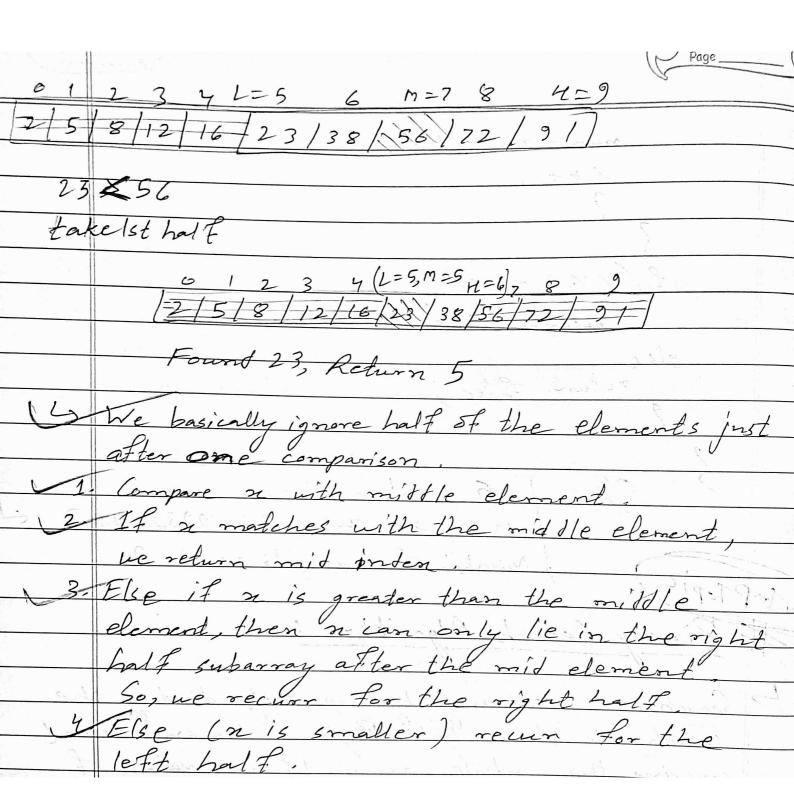
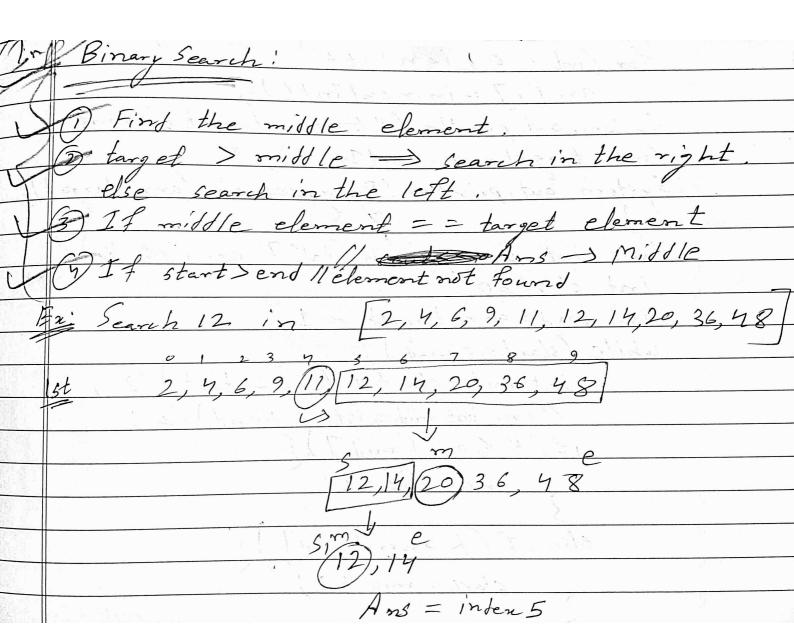
Search Algorithm 138/55 116/23 8/12 Ex 4=9 1=0 take 2nd half





Time Complenity: Best Case: O(1).
Worst Case: O(109m) Worst Case: En! 1180 binary search, freturn the indem of the elements Import java. util. Scanner;

Support java. util. Scanner;

public class Binary Search {

1 (4. in [7] public class Binary Search {

public static void main (String [] args) {

public static void main (String [] args) {

Summer in = new Scanner (System.in)

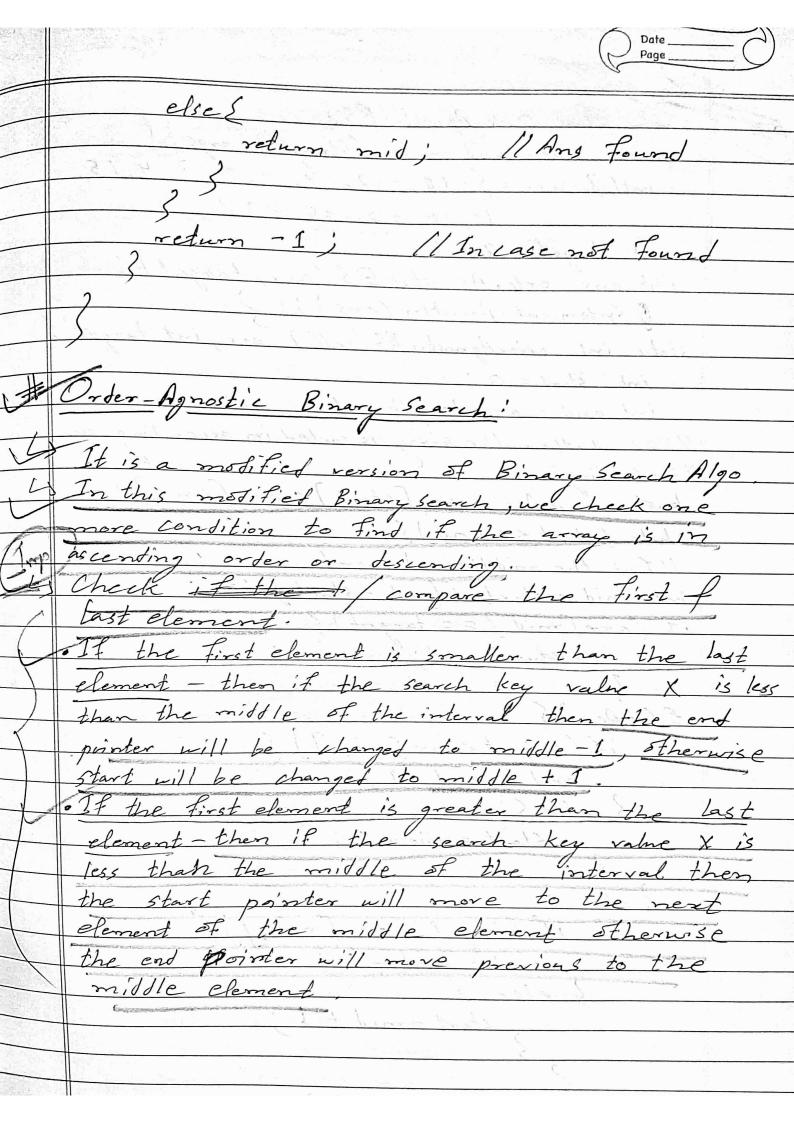
System. out. println [" Enter the size of array!"], int n=in.next Int(); int[] arr = new int [n] System.out.println ("Enter the array!"), For (inti=0; i<n; i++) arr[i] = in next Int(); System. out. println ("Enter target element!");

int target = in. next Int (). System. out. println (Binarysearch (arr, target)); static int Binarysearch (int(7 arr, int K)) int end = arr. length - 1; while (start <= ent) { int mid = start + (end - start)/2;

// or int mid = (start + end)/2;

if (K < arm [m: 17) (if (K < arr [mid]) { 2 end = mid - 1; else if (K) arr [mit]) {

start = mid + 1;



public class Order Agnostic BS & public class Order Hynosticus

public static void main (String[] angs)

int[] arr = [-18, -12, -4, 0, 2, 3, 4, 15, 16, 18, 22, 45, 89 3 int target = 22; int ans = order Agnostic BS (arr, target); System.out. println (ans);

static int orderAgnosticBS (intl] arr, int target) {

ind start = 0; I'm ent = arr, length -1;

If ind whether the array is sorted in ascending on

I tescending boolean is Asc = arr[start] carr[end] while (start <= end) { If find the middle element int mid = start + (end - start)/2; if (arr [mid] = = target) (
return mid: if (is Asc) { if (target Larr [mid]) { end = mid -1; } else { Start = mid + 1; Selse & Selse &

if [target > arr[mid]] {

end = mid - 1; S else S Start = mid + 1; 5 return -1; 33