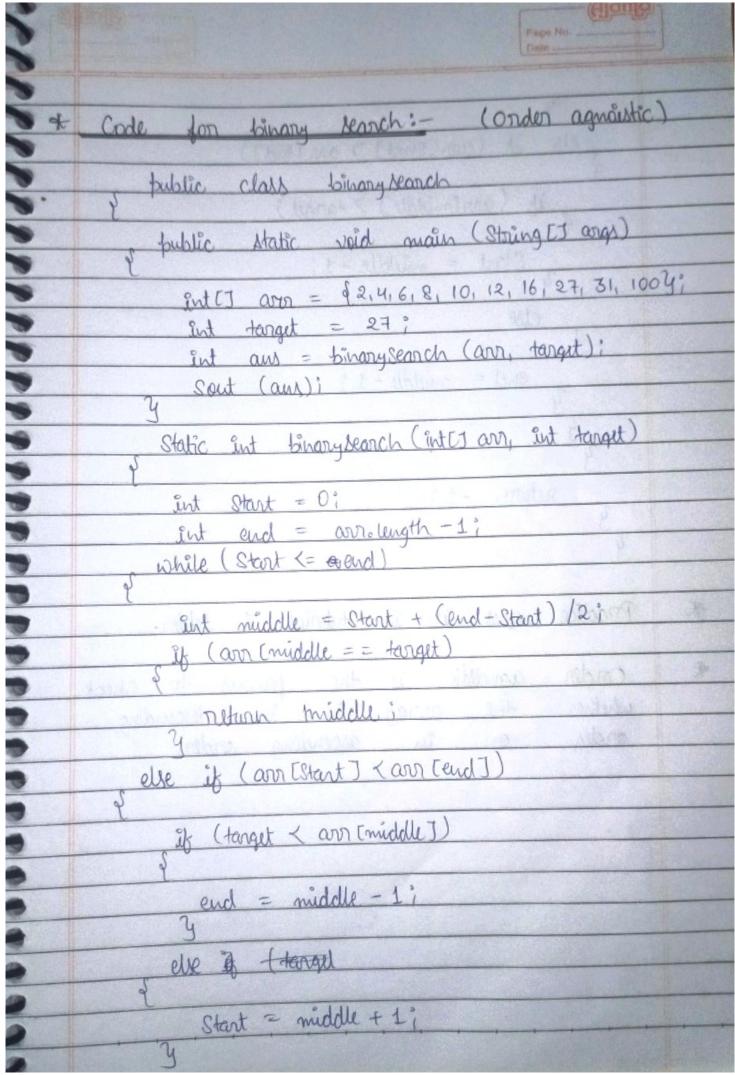
Q>	why binary search? The binary search algorithm was the divide
aus	and conquer approach, it does not scan
	every element in the array, it only reach
	half of the list intead of going though
	each element. hence it is said to be
	best searching algorithm.
	Ex> array of 10,00,000 elements.
	linear search will take 1000000 Comparison in
	worst case. But binary search will take only
	20 comparison. because log 1000000 = 20
	worst case > O(logN).
	Miles de la constant
-	Logic and algorithm: (avray in ascending or
	i) Find the middle index.
	is compare it with target value.
	iii) If target > middle -> Search in the night
	Side
	in the left side.
	v> if target == middle > Sout middle
	000 = [2,4,8,10,(12)14,16,18,20]
	target > 143 middle = St (end
10/20/06/07	Since target > midelle

	So search in the right side.
	now tanget < middle So our end will be shifted. (end=m-1) Start end ann = [2,4,6,8,10,12,49,16,18,20] middle
	Since target == middle Ly Return middle Better way to find middle element:-
•	because int trove some pravae which might
VO (VI	the exceed and we'll be getting enrors. That's why we are using like this.
*	The one-prequisite of binary search is that an array should be in Sorted order. It can be in ascending or descending order.
4500) 4	Total ariger and 20 and
	Albhors Horst 11/2



else its (arm (start) 7 ann [end]) of its (arramiddle) > target) Start = middle +1; else y end = middle - 1; return -1; Practice ques is in tutorial 7 folder. 2 order agnostic is the process to check 头 the array is in descending whether on in ascending order. order The Honding Hard Lymburg roo > trong) - di Horself Brook 1 + alphin - trolo