PAGE NO.:
DATE:

K = 9

0 1 2 3 4
5 6 7 8 9 10 11 12 13 14

[4, 4, 4, 4, 4]

4 < K

6 > K

9 - (arr[h] - (h+1))

9 - 4 = 5

For K^{th} Missing element in an sort array which contains missing elements also

missing element → (2) (5,6) (8,9,10)

9th is the 5th missing element

K = 5

0 1 2 3 4
1 3 4 7 11

[0, 1, 1, 3, 6]

Here we need to find the no of missing element at index by using arr[i] - [i+1]

arr[i] → wrong element in the position

[i+1] → position

use Binary Search

So here two edge cases are decided by the low and high position

Case 1:- if [1, 2, 3, 4] K = 2

0, 0, 0, 0 ← h

So here low = no value as no missing element before the elements or in between

so just return arr[h] + K

here 4 + 2 → 6 value

Case

PAGE NO.:
DATE:

Case 2:- K=9

(1,2,3,4) [5 6 7 8 9]
4 4 4 4 4

so here low value will be at zero index and high will be at n-1
missing same missing
and low == high that means there is no missing element inbetween but precisely there are 4 element missing so here do the following

K-
remaining_missing = [arr[h] - (h+1)]
// 9-4 → 5 missing element

return arr[n-1] + remaining_missing;
// 9+5 → 14 missing element

Case 3:- In between missing element position.

now here l → starting mising position
& h → at the last position of the array.

find mid = (l+h)>>1;
Now if (arr[mid] - (mid+1)) >= K
go to left
store mid

else
go to right

int remain_missing = (arr[res] - (res+1)) - K;
return (arr[res] - 1) - remain_missing;