FIND NUMBER WITH MINIMUM DIFF FROM TARGET

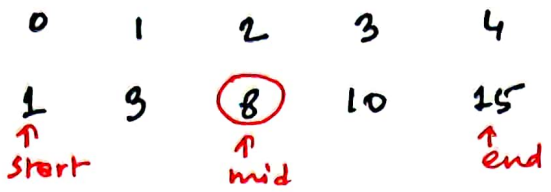
Given an array of numbers in ascending order and a key, find element in the array that has minimum difference with given key.

Example: Input: [4, 6, 10] key → 7
Output: 6

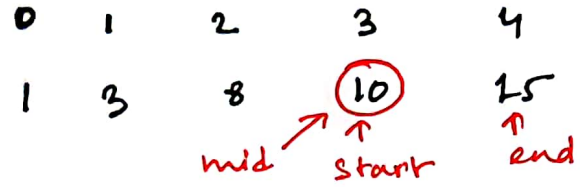
Input: [1, 3, 8, 10, 15] key → 12
Output: 10

Solution:
→ Do a binary search for key in array
→ At each iteration, calculate the current difference: abs(key - A[mid])
→ if this difference is less than difference so far, update min difference to current difference
→ also record mid
→ loop ends if either:
key exists in array, so return key
else, if key not in array
return the element with minimum difference

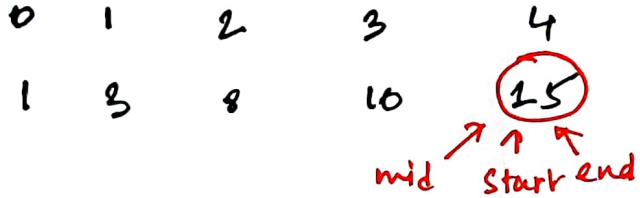
EXAMPLE: [1, 3, 8, 10, 15] key → 12



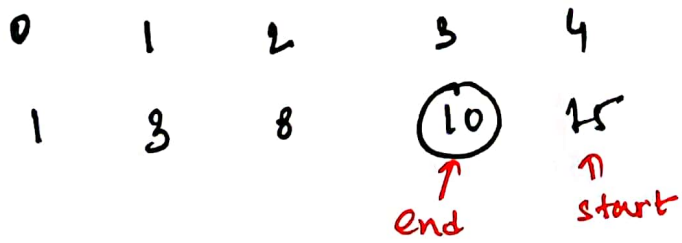
mindiff = maxINT
A[mid] = 8
curr_diff =
abs(A[mid] - key)
= 4
mindiff = 4
result = 8



A[mid] = 10
curr_diff =
abs(10 - 12) = 2
mindiff = min(2, 4)
= 2
result = 10



A[mid] = 15
curr_diff =
abs(12 - 15) = 3
mindiff =
min(2, 3) = 2
result = 10



start > end
loop ends
return 10

OPTIMIZE! if the key < arr[0] :
return arr[0]
if key is > arr[-1]:
return arr[-1]