

164. You are given a sorted array 3,9,14,19,25,31,42,47,53 and asked to find the position of the element 31 using Binary Search. Show the mid-point calculations and the steps involved in finding the element. Display, what would happen if the array was not sorted, how would this impact the performance and correctness of the Binary Search algorithm?

Input : N= 9, a[] = {3,9,14,19,25,31,42,47,53}, search key = 31

Output : 6

Test cases

Input : N= 7, a[] = {13,19,24,29,35,41,42}, search key = 42

Output : 7

Test cases

Input : N= 6, a[] = {20,40,60,80,100,120}, search key = 60

Output : 3

import heapq

PROGRAM :-

import heapq

def kClosest(points, k):

heap = []

for x, y in points:

dist = -(x*x + y*y)

if len(heap) == k:

heapq.heappushpop(heap, (dist, x, y))

else:

heapq.heappush(heap, (dist, x, y))

return [(x, y) for (dist, x, y) in heap]

Test the function

points = [[1, 3], [-2, 2], [5, 8], [0, 1]]

k = 2

print(kClosest(points, k)) # Output: [[-2, 2], [0, 1]]

OUTPUT:-

`[(-2, 2), (0, 1)]`

`=== Code Execution Successful ===`

TIME COMPLEXITY:-O(log N)