Eg: [1, 2, 3, 1] indexDiff = 3

sorted = [ ] valueDiff = 0

→ nums[i] - valueDiff = 1 - 0 = 1

pos = bisectlyt(1) = 0

Is pos < len(sorted) = No.

so insert num[i] in sorted = [1]

→ [1 2 3 1]
num
ptr

nums[i] - valueDiff = 2 - 0 = 2

pos = sorted.bisectleyt(2) = 1

Is pos < len(sorted) = No,

so Insert nums[i] in sorted = [1 2]

→ [1 2 3 1]
num
ptr

nums[i] - valueDiff = 3 - 0 = 3

pos = sorted.bisectleyt(3) = 2

Is pos < len(sorted) = No,

so Insert 3 in sorted = [1, 2, 3]

→ [1 2 3 1] (num)

ph

nums[i] - valueDiff = 1 - 0 = 1

pos = sorted.bisect_left(1) = 0

Now, pos < len(sorted) => yes

$|nums[i] - sorted[pos]| \leq valueDiff$

$= |1 - 1| \leq 0$ = yes

So we return True

Eg: [1 5 9 1], indexDiff = 2, valueDiff = 3

⇒ Sorted = [ ]

num

* [1 5 9 1]

ptr

num - valueDiff

so: 1 - 3 = -2

pos = 0

pos < len(sorted) ? → No

so sorted = [1]

num

* [1 5 9 1]

ptr

so, 5 - 3 = 2

pos = 1

pos < len(sorted) ? → No

so sorted = [1, 5]

num

* [1 5 9 1]

ptr

so 9 - 3 = 6

pos = 2

pos < len(sorted) ? → No

But also len(sorted) = indexDiff, so we remove the

element that was inserted in Start.

So the ptr tell us about the element that was inserted previously.

- remove nums[ptr] = 1, from sorted

sorted = [5, 9]

= move ptr by 1 pos

* [1 5 9 1]
  num
  1
  ptr

so 1 - 3 = -2

pos = 0

But nums[i] - sorted[pos] ≤ valueDiff ?, No,

again len(sorted) = indexDiff, so remove the element

Sorted = [1, 9],

num = [1 5 9 1]
↑
ptr

* End of num → Hence false.