

# Approach

Firstly, we need to understand that for each  $\text{nums}[i]$  we need to find a number which is either equal to it or a number just greater than it, or just smaller than it in the case when  $\text{nums}[i]$  is the greatest among all numbers iterated till now.

Suppose we are currently on  $\text{nums}[i]$ , so to find a number which would have least absolute difference between  $\text{nums}[i]$  and any other number in  $\text{nums}$ , we could greedily choose a number such that the number is just greater or just smaller or equal to  $\text{nums}[i]$ .

So, this could be done with the help of set and lower bound.

Set will help us to keep the sorted order of values iterated till now and then we could easily apply lower bound to it.

The handwritten notes illustrate the approach for finding the minimum absolute difference between a number in an array and its neighbors using a set and lower bound.

**Initial Setup:**

$\text{nums} = [5, 3, 2, 10, 15]$  (indices 0 to 4)  
 $\text{ans} = 1$  and  $1e9$

**Starting from index 1.**

**i=1:**  $\text{nums} = [5, 3, 2, 10, 15]$   
set  $\Rightarrow [5]$   
Lower bound returns index of 5  
Applying lower bound and finding a number equal to  $\text{nums}[i]$  or a number just greater than  $\text{nums}[i]$  in vector  $\text{nums}$ .  
Here, we get  $\text{ans} = \min(1e9, 5 - 3)$   
 $\text{ans} = 2$

**i=2:**  $\text{nums} = [5, 3, 2, 10, 15]$   
set  $\Rightarrow [3, 5]$   
LB returns ind of 3  
 $\text{ans} = 1$

**i=3:**  $\text{nums} = [5, 3, 2, 10, 15]$   
set  $\Rightarrow [2, 3, 5]$   
LB returns pointer at the end so do  $\text{ind} - 1$  to bring it inside the set  
Now, set  $\Rightarrow [2, 3, 5]$   
 $\text{ans} = 1$

**i=4:**  $\text{nums} = [5, 3, 2, 10, 15]$   
set  $\Rightarrow [2, 3, 5, 10]$   
 $\text{ans} = 1$