Problem Discussion:

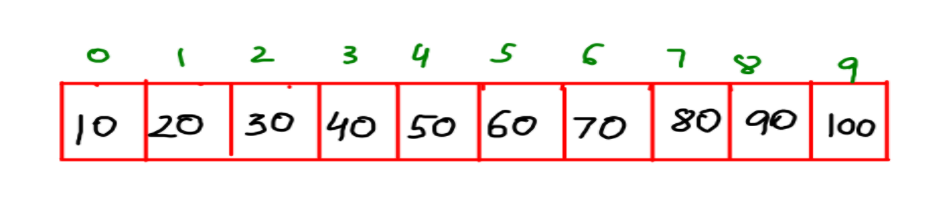
IIf we take a look at the question, we are essentially required to find the **ceil** and **floor** values of the given data, say "d", in a sorted array of n elements.

Now you may ask, what the hell are these ceil and floor values!? No need to be hostile , my friend. Read further.

Ceil value is the least integer that is greater or equal to the given data .Or, **ceil value is the smallest of the elements greater than or equal to the given data "d"**.

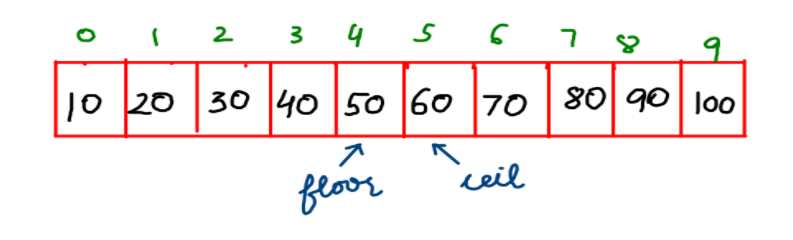
Likewise floor value is the greatest integer that is less than or equal to x. Or, **floor value is the largest of the elements smaller than or equal to the given data "d"**.

It's totally possible that you might not have understood their meaning entirely ,so let's discuss a few examples.

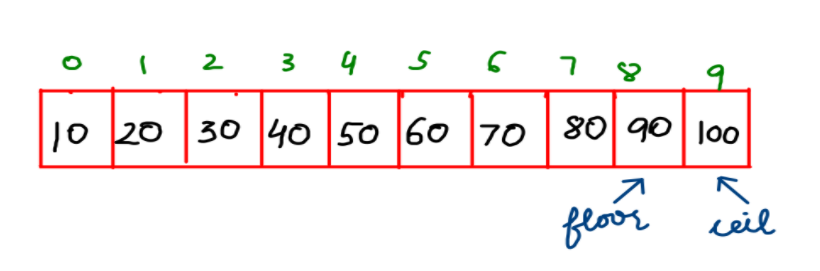


We are given the above sorted array arr[ ] of 10 elements.

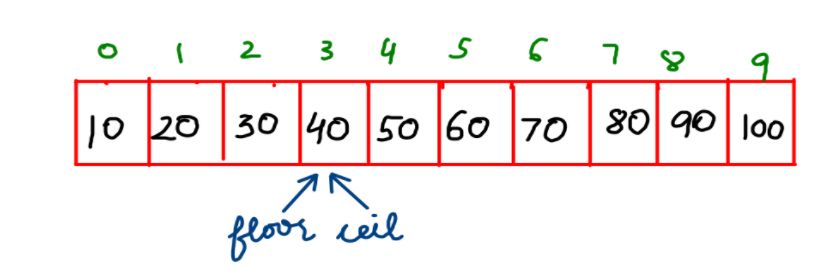
**Example 1 :**d=55



**Example 2 :**d=92



**Example 3 :**d= 40 , Here, since 40 is an element in the array, therefore ceil and floor are both equal to 40.



Approach:

Let's discuss the strategy to find the ceil and floor values of d=33. Pay close attention to it !

First, we initialize floor value as -8 and ceil value as +8. Also, "lo" is initialized as 0 and "hi" as arr.length-1, as we have already examined in the Binary Search Algorithm.

1. Then, by the Binary Search algorithm, mid =4 of value 50. Since 33<50, therefore hi=mid-1= 3 .Hence, we set the ceil as the mid value 50.
2. Now , for lo=0 and hi=3, mid=1 of value 20. Since 33>20, therefore lo=mid+1=2. Hence, we set the floor as the mid value 20.
3. By now, lo=2 and hi=3 , so mid =2 of value 30 , Since 33>30, therefore lo=mid+1=3. Hence, we set the floor as the mid value 30.
4. Now, since lo and hi both are at index 3, so mid=3 of value 40.Since,33<40, therefore hi=mid-1= 2 .Hence, we set the ceil as the mid value 40.
5. Since lo and hi have crossed each other, we'll terminate the loop and return the values of ceil and floor.

We highly request you to refer to the Youtube video "Ceil and Floor" for a clear picture of how this program should run.

If ceil and floor are to be found for "d" ,

Then three cases arise.

**Case I :** d< arr[mid] ?? hi=mid-1 , ceil=arr[mid]

**Case II :** d>arr[mid] ??lo=mid+1, floor=arr[mid]

**Case III :** d=arr[mid] ?? ceil=arr[mid] ,floor=arr[mid] , break

Analysis

**Time Complexity:**

**O(logn)**

This time complexity is logarithmic since a binary search-like approach is used and the search interval repeatedly gets reduced by half.

**SPACE COMPLEXITY:**

**O(1)**

Since no extra space is required for solving the problem therefore the space complexity is constant.