Ceiling in a sorted array

Given a sorted array and a value x, the ceiling of x is the smallest element in an array greater than or equal to x, and the floor is the greatest element smaller than or equal to x. Assume that the array is sorted in non-decreasing order. Write efficient functions to find the floor and ceiling of x.

Examples :

For example, let the input array be {1, 2, 8, 10, 10, 12, 19}

For x = 0: floor doesn't exist in array, ceil = 1

For x = 1: floor = 1, ceil = 1

For x = 5: floor = 2, ceil = 8

For x = 20: floor = 19, ceil doesn't exist in array

In the below methods, we have implemented only ceiling search functions. Floor search can be implemented in the same way.

Method 1 (Linear Search)

Algorithm to search ceiling of x:

1. If x is smaller than or equal to the first element in the array then return 0(index of the first element).
2. Else linearly search for an index i such that x lies between arr[i] and arr[i+1].
3. If we do not find an index i in step 2, then return -1.