

Day-2 Challenge Arsh Goyal

Ceil The Floor

Basic

Accuracy: 43.76%

Submissions: 31K+

Points: 1

Sharpen up your programming skills, participate in coding contests & explore high-paying jobs

Given an unsorted array `Arr[]` of `N` integers and an integer `X`, find floor and ceiling of `X` in `Arr[0..N-1]`.

Floor of `X` is the largest element which is smaller than or equal to `X`. Floor of `X` doesn't exist if `X` is smaller than smallest element of `Arr[]`.

Ceil of `X` is the smallest element which is greater than or equal to `X`. Ceil of `X` doesn't exist if `X` is greater than greatest element of `Arr[]`.

Example 1:

Input:

`N = 8, X = 7`

`Arr[] = {5, 6, 8, 9, 6, 5, 5, 6}`

Output: 6 8

Explanation:

Floor of 7 is 6 and ceil of 7 is 8.

we are given with `X` which will help us to find
Ciel and floor of from array. ~~As~~ we have unsorted Array
so we will sort it first

Floor \rightarrow largest element in Array but $\leq x$

Ciel \rightarrow minimum element in Array but $\geq x$

Example:- arr = [10, 20, 30, 40, 50]

$x = 25$

Ciel =
 $50 \geq 25$
 $40 \geq 25$
 $30 \geq 25 \rightarrow \text{Ciel}$
 $20 \geq 25$ False

Floor = $10 \leq 25$
 $20 \leq 25 \rightarrow \text{Floor}$
 $30 \leq 25$ False

arr = [10, 20, 25, 30, 35, 40, 50]

Ciel =
 $50 \geq 25$
 $40 \geq 25$
 $35 \geq 25$
 $30 \geq 25$
 $25 \geq 25 \rightarrow \text{Ciel}$
 $20 \geq 25$ False

(lower bound)

Floor = $10 \leq 25$
 $20 \leq 25$
 $25 \leq 25 \rightarrow \text{Floor}$
 $30 \leq 25$ False

(upper bound)

Upper bound : Floor $\xrightarrow{\text{Floor of } x}$ $\xrightarrow{\text{size}}$
`int floorX (int arr[], int x, int n)`

`ans = -1`

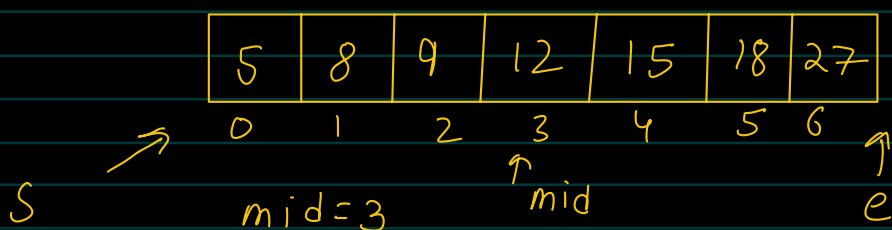
`int s = 0, end = n - 1;`

`while (s <= e)`

`int mid = s + (e - s) / 2;`

`if (arr[mid] <= x)`

$x = 16$

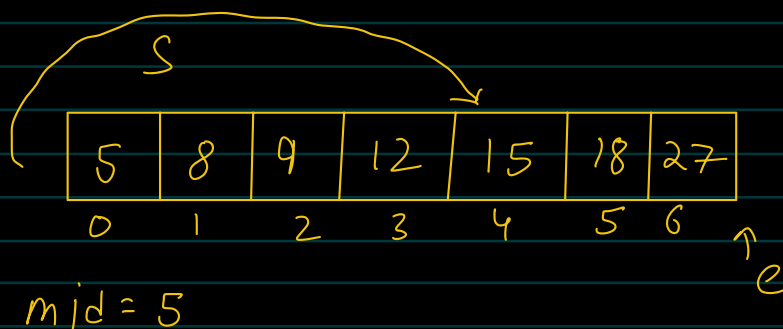


$12 \leq 16$ True

Save your `arr[mid]` in variable

Since we have find largest possible and Array is sorted so element Right hand side of mid will be larger than `arr[mid]`

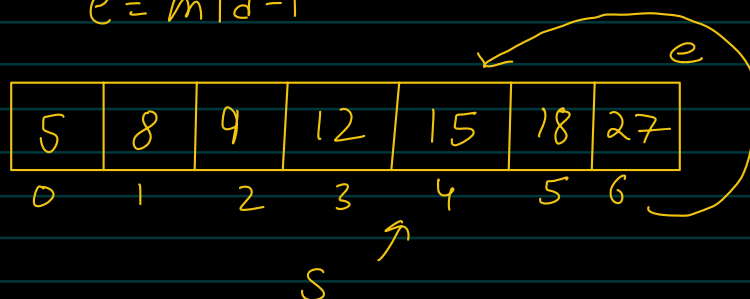
$s = \text{mid} + 1$, $\text{Ans} = 12$



`if (18 <= 16)` False

Reduce from e

$e = \text{mid} - 1$



$\text{mid} = 4$, `if (15 <= 16)` True
 $\text{Ans} = 15$

				S		
				↖	↘	
5	8	9	12	15	18	27
0	1	2	3	4	5	6
			e			

while (S ≤ e) False

return ans;

Similarly for c1d

if (arr[mid] ≥ x)

ans = arr[mid]

Since minimum possible answer

e = mid - 1

Code

```
int floorX(int arr[],int n,int x){
    int s=0;
    int e=n-1;
    int ans=-1;
    while(s<=e){
        int mid=s+(e-s)/2;
        if(arr[mid]<=x){
            ans=arr[mid];
            s=mid+1;
        }
        else{
            e=mid-1;
        }
    }
    return ans;
}

int CielX(int arr[],int n,int x){
    int s=0;
    int e=n-1;
    int ans=-1;
    while(s<=e){
        int mid=s+(e-s)/2;
        if(arr[mid]>=x){
            ans=arr[mid];
            e=mid-1;
        }
        else{
            s=mid+1;
        }
    }
    return ans;
}

pair<int, int> getFloorAndCeil(int arr[], int n, int x) {
    sort(arr,arr+n);
    int floorValue=floorX(arr,n,x);
    int CielValue=CielX(arr,n,x);
    return make_pair(floorValue,CielValue);
}
```