

# lower\_bound in C++

Difficulty Level : Easy • Last Updated : 28 Oct, 2020

The **lower\_bound()** method in C++ is used to return an iterator pointing to the first element in the range [first, last) which has a value not less than val. This means that the function returns the index of the next smallest number just greater than or equal to that number. If there are multiple values that are equal to val, lower\_bound() returns the index of the first such value.

The elements in the range shall already be sorted or at least partitioned with respect to val.

## Templates:

### Syntax 1:

*ForwardIterator lower\_bound (ForwardIterator first, ForwardIterator last, const T& val);*

### Syntax 2:

*ForwardIterator lower\_bound (ForwardIterator first, ForwardIterator last, const T& val, Compare comp);*



**Parameters:** The above methods accept the following parameters.

- **first, last:** The range used is [first, last), which contains all the

elements between first and last, including the element pointed by first but not the element pointed by last.

- **val:** Value of the lower bound to be searched for in the range.
- **comp:** Binary function that accepts two arguments (the first of the type pointed by ForwardIterator, and the second, always val), and returns a value convertible to bool. The function shall not modify any of its arguments. This can either be a function pointer or a function object.

**Return Value:** An iterator to the lower bound of val in the range. If all the elements in the range compare less than val, the function returns last. If all the elements in the range are larger than val, the function returns a pointer to the first element.

### Examples:

**Input:** 10 20 30 40 50

**Output:** lower\_bound for element 30 at index 2

**Input:** 10 20 30 40 50

**Output:** lower\_bound for element 35 at index 3

**Input:** 10 20 30 40 50

**Output:** lower\_bound for element 55 at index 5

**Input:** 10 20 30 30 30 40 50

**Output:** lower\_bound for element 30 at index 2

---

## CPP

```
// CPP program to illustrate
// std :: lower_bound
#include <bits/stdc++.h>

// Driver code
int main()
```



```
{
    // Input vector
    std::vector<int> v{ 10, 20, 30, 30, 30, 40, 50 };

    // Print vector
    std::cout << "Vector contains :";
    for (unsigned int i = 0; i < v.size(); i++)
        std::cout << " " << v[i];
    std::cout << "\n";

    std::vector<int>::iterator low1, low2, low3;

    // std :: lower_bound
    low1 = std::lower_bound(v.begin(), v.end(), 30);
    low2 = std::lower_bound(v.begin(), v.end(), 35);
    low3 = std::lower_bound(v.begin(), v.end(), 55);

    // Printing the lower bounds
    std::cout
        << "\nlower_bound for element 30 at position : "
        << (low1 - v.begin());
    std::cout
        << "\nlower_bound for element 35 at position : "
        << (low2 - v.begin());
    std::cout
        << "\nlower_bound for element 55 at position : "
        << (low3 - v.begin());

    return 0;
}
```

## Output:

Vector contains : 10 20 30 30 30 40 50

lower\_bound for element 30 at position : 2

lower\_bound for element 35 at position : 5

lower\_bound for element 55 at position : 7



**Time Complexity:** The number of comparisons performed is logarithmic. Therefore, the time complexity of the above approach is  **$O(\log N)$** , where  $N = \text{size} \cdot (\text{last} - \text{first})$

This article is contributed by [Sachin Bisht](#). If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://contribute.geeksforgeeks.org) or mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Want to learn from the best curated videos and practice problems, check out the [C++ Foundation Course](#) for Basic to Advanced C++ and [C++ STL Course](#) for foundation plus STL. To complete your preparation from learning a language to DS Algo and many more, please refer [Complete Interview Preparation Course](#).



**Like** 71

Previous

**std::upper\_bound and  
std::lower\_bound for Vector  
in C++ STL**

Next

**upper\_bound in C++**



## RECOMMENDED ARTICLES

Page : 1 2

- 01** What will happen if a `print()` statement is written inside a `if()` such as `if(print())`  
13, Aug 21
- 02** How to find Size of `std::forward_list` in C++ STL  
13, Aug 21
- 03** Draw a moving cycle using computer graphics programming in C/C++  
01, Aug 21
- 04** Abnormal behavior of floating point and double values  
01, Aug 21
- 05** Difference between long int and long long int in C/C++  
01, Aug 21
- 06** Array of Sets in C++ STL  
30, Jul 21
- 07** Print Adjacency List for a Directed Graph  
29, Jul 21
- 08** Applications and Popularities of C++  
29, Jul 21

## Article Contributed By :



## Vote for difficulty

Current difficulty : [Easy](#)

Easy

Normal

Medium

Hard

Expert

**Improved By :** [abhishek\\_987](#), [theWINTERSOLDIER](#), [shruti10gandotra](#),  
[mayukhchakrabarti99](#), [aaaayush25](#)

**Article Tags :** [cpp-algorithm-library](#), [cpp-binary-search](#), [STL](#), [C++](#)

**Practice Tags :** [STL](#), [CPP](#)

[Improve Article](#)[Report Issue](#)[About Us](#)[Careers](#)[Privacy Policy](#)[Contact Us](#)[Copyright Policy](#)

## Practice

[Courses](#)[Company-wise](#)[Topic-wise](#)[How to begin?](#)[Algorithms](#)[Data Structures](#)[Languages](#)[CS Subjects](#)[Video Tutorials](#)

## Contribute

[Write an Article](#)[Write Interview Experience](#)[Internships](#)[Videos](#)

@geeksforgeeks , Some rights reserved

