

List Built-in Functions:

1. Constructor

| Name | Details | Time Complexity |
|---|---|-----------------|
| <code>list<type>myList;</code> | Construct a list with 0 elements. | O(1) |
| <code>list<type>myList(N);</code> | Construct a list with N elements and the value will be garbage. | O(N) |
| <code>list<type>myList(N,V);</code> | Construct a list with N elements and the value will be V. | O(N) |
| <code>list<type>myList(list2);</code> | Construct a list by copying another list list2. | O(N) |
| <code>list<type>myList(A,A+N);</code> | Construct a list by copying all elements from an array A of size N. | O(N) |

2. Capacity

| Name | Details | Time Complexity |
|--------------------------------|--|--|
| <code>myList.size()</code> | Returns the size of the list. | O(1) |
| <code>myList.max_size()</code> | Returns the maximum size that the vector can hold. | O(1) |
| <code>myList.clear()</code> | Clears the list elements. Do not delete the memory, only clear the list. | O(N) |
| <code>myList.empty()</code> | Return true/false if the list is empty or not. | O(1) |
| <code>myList.resize()</code> | Change the size of the list. | O(K); where K is the difference between new size and current size. |

3. Modifiers

| Name | Details | Time Complexity |
|--|---|---|
| myList= or myList.assign(list 2.begin(),list2.end()) | Assign another list. | O(N) |
| myList.push_back () | Add an element to the tail. | O(1) |
| myList.push_front () | Add an element to the head. | O(1) |
| myList.pop_back() | Delete the tail. | O(1) |
| myList.pop_front() | Delete the head. | O(1) |
| myList.insert() | Insert elements at a specific position. | O(N+K); where K is the number of elements to be inserted. |
| myList.erase() | Delete elements from a specific position. | O(N+K); where K is the number of elements to be deleted. |
| replace(myList.be gin(),myList.end(), value,replace_val ue) | Replace all the value with replace_value. Not under a list STL. | O(N) |
| find(myList.begin() ,myList.end(),V) | Find the value V. Not under a list STL. | O(N) |

4. Operations

| Name | Details | Time Complexity |
|-------------------------|-------------------------|-----------------|
| myList.remove(V) | Remove the value V from | O(N) |

| | | |
|---|---|----------------------------------|
| | the list. | |
| myList.sort() | Sort the list in ascending order. | $O(N \log N)$ |
| myList.sort(greater<type>()) | Sort the list in descending order | $O(N \log N)$ |
| myList.unique() | Deletes the duplicate values from the list. You must sort the list first. | $O(N)$, with sort $O(N \log N)$ |
| myList.reverse() | Reverse the list. | $O(N)$ |

5. Element access

| Name | Details | Time Complexity |
|--------------------------------|--------------------------------|-----------------|
| myList.back() | Access the tail element. | $O(1)$ |
| myList.front() | Access the head element. | $O(1)$ |
| next(myList.begin(), i) | Access the <i>i</i> th element | $O(N)$ |

6. Iterators

| Name | Details | Time Complexity |
|-----------------------|-------------------------------|-----------------|
| myList.begin() | Pointer to the first element. | $O(1)$ |
| myList.end() | Pointer to the last element. | $O(1)$ |