Introduction to Basic Data Structures

List Built-in Functions:

1. Constructor

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

2. Capacity

Name	Details	Time Complexity
myList.size()	Returns the size of the list.	O(1)
myList.max_size()	Returns the maximum size that the list can hold.	O(1)

myList.clear()	Clears the list elements.	O(N)
myList.empty()	Return true/false if the list is empty or not.	O(1)
myList.resize()	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(list2.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.begin(),myList.end(),value,replace_value)	Replace all the value with replace_value. Not under a list STL.	O(N)
find(myList.begin(	Find the value V. Not under	O(N)

),myList.end(),V)	a list STL.	

4. **Operations**

Name	**Details**	**Time Complexity**
myList.remove(V)	Remove the value V from the list.	O(N)
myList.sort()	Sort the list in ascending order.	O(NlogN)
myList.sort(greater<type>())	Sort the list in descending order	O(NlogN)
myList.unique()	Deletes the duplicate values from the list. You must sort the list first.	O(N), with sort O(NlogN)
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 ith 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)