Vector Built-in Functions:

1. **Initialization**

Name	**Details**	**Time Complexity**
vector<type>v;	Construct a vector with 0 elements.	O(1)
vector<type>v(N);	Construct a vector with N elements.	O(N)
vector<type>v(N,V);	Construct a vector with N elements and the value will be V.	O(N)
vector<type>v(v2);	Construct a vector by copying another vector v2.	O(N)
vector<type>v(A,A+N);	Construct a vector by copying all elements from an array A of size N.	O(N)

2. **Capacity**

Name	**Details**	**Time Complexity**
v.size()	Returns the size of the vector.	O(1)
v.max_size()	Returns the maximum size that the vector can hold.	O(1)
v.capacity()	Returns the current available capacity of the vector.	O(1)
v.clear()	Clears the vector elements. Do not delete the memory, only clear the value.	O(N)
v.empty()	Return true/false if the vector is empty or not.	O(1)
v.resize()	Change the size of the vector.	O(K); where K is the difference between new size and current size.

3. **Modifiers**

Name	**Details**	**Time Complexity**
v= or v.assign()	Assign another vector.	O(N) if sizes are different, O(1) otherwise.
v.push_back()	Add an element to the end.	O(1)
v.pop_back()	Remove the last element.	O(1)
v.insert()	Insert elements at a specific position.	O(N+K); where K is the number of elements to be inserted.
v.erase()	Delete elements from a specific position.	O(N+K); where K is the number of elements to be deleted.
replace(v.begin(),v.end(),value,replace_value)	Replace all the value with replace_value. Not under a vector.	O(N)
find(v.begin(),v.end(),V)	Find the value V. Not under a vector.	O(N)

4. **Element access**

Name	**Details**	**Time Complexity**
v[i]	Access the ith element.	O(1)
v.at(i)	Access the ith element.	O(1)
v.back()	Access the last element.	O(1)
v.front()	Access the first element.	O(1)

5. Iterators

Name	**Details**	**Time Complexity**
v.begin()	Pointer to the first element.	O(1)
v.end()	Pointer to the last element.	O(1)