

Deque Built-in Functions in C++ STL

1. Constructor

Name	Details	Time Complexity
<code>deque<type> d;</code>	Construct a deque with 0 elements.	O(1)
<code>deque<type> d(N);</code>	Construct a deque with N elements and garbage values.	O(N)
<code>deque<type> d(N, V);</code>	Construct a deque with N elements, each of value V.	O(N)
<code>deque<type> d(d2);</code>	Construct a deque by copying another deque d2.	O(N)
<code>deque<type> d(A, A+N);</code>	Construct a deque by copying all elements from an array A of size N.	O(N)

2. Capacity

Name	Details	Time Complexity
<code>d.size()</code>	Returns the number of elements in the deque.	O(1)
<code>d.max_size()</code>	Returns the maximum size the deque can hold.	O(1)
<code>d.empty()</code>	Returns true if the deque is empty, otherwise false.	O(1)
<code>d.resize(N)</code>	Resizes the deque to contain N elements.	O(N)
<code>d.shrink_to_fit()</code>	Reduces memory usage by freeing unused memory.	O(N)

3. Modifiers

Name	Details	Time Complexity
<code>d.push_back(V)</code>	Adds an element V to the end of the deque.	O(1)
<code>d.push_front(V)</code>	Adds an element V to the front of the deque.	O(1)
<code>d.pop_back()</code>	Removes the last element of the deque.	O(1)
<code>d.pop_front()</code>	Removes the first element of the deque.	O(1)
<code>d.insert(pos, V)</code>	Inserts element V at position pos.	O(N)
<code>d.erase(pos)</code>	Removes element at position pos.	O(N)
<code>d.clear()</code>	Clears all elements from the deque.	O(N)
<code>d.assign(N, V)</code>	Assigns N elements of value V to the deque.	O(N)
<code>d.swap(d2)</code>	Swaps contents with another deque d2.	O(1)

4. Element Access

Name	Details	Time Complexity
<code>d[i]</code>	Accesses the <i>i</i> th element.	$O(1)$
<code>d.at(i)</code>	Accesses the <i>i</i> th element with bounds checking.	$O(1)$
<code>d.front()</code>	Accesses the first element.	$O(1)$
<code>d.back()</code>	Accesses the last element.	$O(1)$

5. Iterators

Name	Details	Time Complexity
<code>d.begin()</code>	Returns an iterator to the first element.	$O(1)$
<code>d.end()</code>	Returns an iterator to the element following the last element.	$O(1)$
<code>d.rbegin()</code>	Returns a reverse iterator to the last element.	$O(1)$
<code>d.rend()</code>	Returns a reverse iterator to the element preceding the first element.	$O(1)$