

			Sequence containers			Associative containers				
Headers			<vector>	<deque>	<list>	<set>	<map>		<bitset>	
Members		complex	vector	deque	list	set	multiset	map	multimap	bitset
	<i>constructor</i>	*	constructor	constructor	constructor	constructor	constructor	constructor	constructor	constructor
	<i>destructor</i>	O(n)	destructor	destructor	destructor	destructor	destructor	destructor	destructor	
	<i>operator=</i>	O(n)	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operators
iterators	<i>begin</i>	O(1)	begin	begin	begin	begin	begin	begin	begin	
	<i>end</i>	O(1)	end	end	end	end	end	end	end	
	<i>rbegin</i>	O(1)	rbegin	rbegin	rbegin	rbegin	rbegin	rbegin	rbegin	
	<i>rend</i>	O(1)	rend	rend	rend	rend	rend	rend	rend	
capacity	<i>size</i>	*	size	size	size	size	size	size	size	size
	<i>max_size</i>	*	max_size	max_size	max_size	max_size	max_size	max_size	max_size	
	<i>empty</i>	O(1)	empty	empty	empty	empty	empty	empty	empty	
	<i>resize</i>	O(n)	resize	resize	resize					
element access	<i>front</i>	O(1)	front	front	front					
	<i>back</i>	O(1)	back	back	back					
	<i>operator[]</i>	*	operator[]	operator[]				operator[]		operator[]
	<i>at</i>	O(1)	at	at						
modifiers	<i>assign</i>	O(n)	assign	assign	assign					
	<i>insert</i>	*	insert	insert	insert	insert	insert	insert	insert	
	<i>erase</i>	*	erase	erase	erase	erase	erase	erase	erase	
	<i>swap</i>	O(1)	swap	swap	swap	swap	swap	swap	swap	
	<i>clear</i>	O(n)	clear	clear	clear	clear	clear	clear	clear	
	<i>push_front</i>	O(1)		push_front	push_front					
	<i>pop_front</i>	O(1)		pop_front	pop_front					
	<i>push_back</i>	O(1)	push_back	push_back	push_back					
observers	<i>pop_back</i>	O(1)	pop_back	pop_back	pop_back					
	<i>key_comp</i>	O(1)				key_comp	key_comp	key_comp	key_comp	
	<i>value_comp</i>	O(1)				value_comp	value_comp	value_comp	value_comp	
operations	<i>find</i>	O(log n)				find	find	find	find	
	<i>count</i>	O(log n)				count	count	count	count	count
	<i>lower_bound</i>	O(log n)				lower_bound	lower_bound	lower_bound	lower_bound	
	<i>upper_bound</i>	O(log n)				upper_bound	upper_bound	upper_bound	upper_bound	
	<i>equal_range</i>	O(log n)				equal_range	equal_range	equal_range	equal_range	
<i>unique members</i>			capacity reserve		splice remove remove_if unique merge sort reverse					set reset flip to_ulong to_string test any none

Amortized complexity shown. Legend: O(1) constant < O(log n) logarithmic < O(n) linear; *=depends on container

Container adaptors:

		Container Adaptors			
Headers		<stack>	<queue>		
Members		stack	queue	priority_queue	
	<i>constructor</i> *	constructor	constructor	constructor	
capacity	size	O(1) size	size	size	
	empty	O(1) empty	empty	empty	
	front	O(1)	front		
element access	back	O(1)	back		
	top	O(1) top		top	
modifiers	push	O(1) push	push	push	
	pop	O(1) pop	pop	pop	