

core::tree

```
template <class T, class Allocator = allocator<T> > class tree;
```

Member types

member type	definition	notes
value_type	The first template parameter (T)	
allocator_type	The second template parameter (Allocator)	defaults to: allocator<T>
reference	value_type&	
const_reference	const value_type&	
pointer	value_type*	
const_pointer	const value_type*	
iterator	a bidirectional iterator to value_type	convertible to: const_iterator
const_iterator	a bidirectional iterator to const value_type	
reverse_iterator	a bidirectional reverse iterator to value_type	
const_reverse_iterator	a bidirectional reverse iterator to const value_type	
primitive_iterator	a bidirectional primitive iterator to value_type	convertible to: const_primitive_iterator
const_primitive_iterator	a bidirectional primitive iterator to const value_type	
reverse_primitive_iterator	a bidirectional reverse primitive iterator to value_type	
const_reverse_primitive_iterator	a bidirectional reverse primitive iterator to const value_type	
sibling_iterator	a bidirectional sibling iterator to value_type	convertible to: const_sibling_iterator

<code>const_sibling_iterator</code>	a bidirectional sibling iterator to <code>const value_type</code>	
<code>reverse_sibling_iterator</code>	a bidirectional reverse sibling iterator to <code>value_type</code>	
<code>const_reverse_sibling_iterator</code>	a bidirectional reverse sibling iterator to <code>const value_type</code>	
<code>leaf_iterator</code>	a bidirectional leaf iterator to <code>value_type</code>	convertible to: <code>const_leaf_iterator</code>
<code>const_leaf_iterator</code>	a bidirectional leaf iterator to <code>const value_type</code>	
<code>reverse_leaf_iterator</code>	a bidirectional reverse leaf iterator to <code>value_type</code>	
<code>const_reverse_leaf_iterator</code>	a bidirectional reverse leaf iterator to <code>const value_type</code>	
<code>size_type</code>	an unsigned integral type that can represent any non-negative value of <code>difference_type</code>	usually the same as <code>size_t</code>
<code>difference_type</code>	a signed integral type	usually the same as <code>ptrdiff_t</code>

Member functions

<code>(constructor)</code>	Construct tree (<code>public member function</code>)
<code>(destructor)</code>	Tree destructor (<code>public member function</code>)
<code>operator=</code>	Assign content (<code>public member function</code>)

Iterators:

<code>begin</code>	Return iterator to beginning (<code>public member function</code>)
<code>end</code>	Return iterator to end (<code>public member function</code>)
<code>rbegin</code>	Return <code>reverse_iterator</code> to reverse beginning (<code>public member function</code>)
<code>rend</code>	Return <code>reverse_iterator</code> to reverse end (<code>public member function</code>)
<code>cbegin</code>	Return <code>const_iterator</code> to beginning (<code>public member function</code>)

<code>cend</code>	Return <code>const_iterator</code> to end (public member function)
<code>crbegin</code>	Return <code>const_reverse_iterator</code> to reverse beginning (public member function)
<code>crend</code>	Return <code>const_reverse_iterator</code> to reverse end (public member function)
<code>pbegin</code>	Return <code>primitive_iterator</code> to beginning (public member function)
<code>pend</code>	Return <code>primitive_iterator</code> to end (public member function)
<code>rpbegin</code>	Return <code>reverse_primitive_iterator</code> to reverse beginning (public member function)
<code>rpend</code>	Return <code>reverse_primitive_iterator</code> to reverse end (public member function)
<code>cbegin</code>	Return <code>const_primitive_iterator</code> to beginning (public member function)
<code>cpend</code>	Return <code>const_primitive_iterator</code> to end (public member function)
<code>crpbegin</code>	Return <code>const_reverse_primitive_iterator</code> to reverse beginning (public member function)
<code>crpend</code>	Return <code>const_reverse_primitive_iterator</code> to reverse end (public member function)
<code>sbegin</code>	Return <code>sibling_iterator</code> to beginning (public member function)
<code>send</code>	Return <code>sibling_iterator</code> to end (public member function)
<code>rsbegin</code>	Return <code>reverse_sibling_iterator</code> to reverse beginning (public member function)
<code>rsend</code>	Return <code>reverse_sibling_iterator</code> to reverse end (public member function)
<code>csbegin</code>	Return <code>const_sibling_iterator</code> to beginning (public member function)
<code>csend</code>	Return <code>const_sibling_iterator</code> to end (public member function)
<code>crsbegin</code>	Return <code>const_reverse_sibling_iterator</code> to reverse beginning (public member function)
<code>crsend</code>	Return <code>const_reverse_sibling_iterator</code> to reverse end (public member function)
<code>lbegin</code>	Return <code>leaf_iterator</code> to beginning (public member function)
<code>lend</code>	Return <code>leaf_iterator</code> to end (public member function)
<code>rlbegin</code>	Return <code>reverse_leaf_iterator</code> to reverse beginning (public member function)
<code>rlend</code>	Return <code>reverse_leaf_iterator</code> to reverse end (public member function)
<code>clbegin</code>	Return <code>const_leaf_iterator</code> to beginning (public member function)
<code>clend</code>	Return <code>const_leaf_iterator</code> to end (public member function)

crlbegin	Return const_reverse_leaf_iterator to reverse beginning (public member function)
crlend	Return const_reverse_leaf_iterator to reverse end (public member function)

Capacity:

empty	Test whether container is empty (public member function)
size	Return size (public member function)
max_size	Return maximum size (public member function)

Modifiers:

assign	Assign new content to container (public member function)
emplace_front	Construct and insert an element at beginning (public member function)
push_front	Insert elements at beginning (public member function)
pop_front	Delete the first element (public member function)
emplace_back	Construct and insert an element at the end (public member function)
push_back	Insert elements at the end (public member function)
pop_back	Delete the last element (public member function)
emplace_child_front	Construct and insert a child node at beginning (public member function)
emplace_child_back	Construct and insert a child node at the end (public member function)
prepend_child	Add child nodes at beginning (public member function)
append_child	Add child nodes at the end (public member function)
emplace	Construct and insert a element (public member function)
insert	Insert elements (public member function)
copy_child_front	Copy child nodes at beginning (public member function)
copy_child_back	Copy child nodes at the end (public member function)
copy_node	Copy nodes (public member function)
erase	Erase elements (public member function)
remove_children	Remove all child nodes (public member function)

swap	Swap content (public member function)
clear	Clear content (public member function)

Operations:

splice	Transfer elements from node to node (public member function)
merge	Merge sorted nodes (public member function)
sort	Sort elements in container (public member function)

Observers:

get_allocator	Get allocator (public member function)
---------------	--