Skip List

Generated by Doxygen 1.9.1

1 Class Index 1

1.1 Class List 1

2 Class Documentation 3

2.1 skip_list< T, Compare, Allocator >::const_iterator Class Reference 3

2.1.1 Detailed Description 4

2.2 skip_list< T, Compare, Allocator >::iterator Class Reference 4

2.2.1 Detailed Description 5

2.3 skip_list< T, Compare, Allocator > Class Template Reference 5

2.3.1 Detailed Description 7

2.3.2 Member Function Documentation 7

2.3.2.1 contains() 7

2.3.2.2 emplace() 7

2.3.2.3 insert() [1/2] 8

2.3.2.4 insert() [2/2] 8

2.3.2.5 push_back() 9

2.3.2.6 push_front() 9

2.3.2.7 resize() 9

2.3.2.8 swap() 10

Index 11

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

skip_list< T, Compare, Allocator >::const_iterator
Constant bidirectional iterator for skip_list 3
skip_list< T, Compare, Allocator >::iterator
Bidirectional iterator for skip_list 4
skip_list< T, Compare, Allocator >
A skip list container that provides logarithmic time complexity for search operations 5

Chapter 2

Class Documentation

2.1 skip_list< T, Compare, Allocator >::const_iterator Class Reference

Constant bidirectional iterator for skip_list.

`#include <SkipList.h>`

Public Types

- using **iterator_category** = std::bidirectional_iterator_tag
- using **value_type** = const T
- using **difference_type** = std::ptrdiff_t
- using **pointer** = const T *
- using **reference** = const T &

Public Member Functions

- **const_iterator** (skip_list_node *node)
- **const_iterator** (const iterator &it)
- reference operator* () const

  Dereference operator.
- pointer operator-> () const

  Member access operator.
- const_iterator & operator++ ()

  Prefix increment.
- const_iterator operator++ (int)

  Postfix increment.
- const_iterator & operator-- ()

  Prefix decrement.
- const_iterator operator-- (int)

  Postfix decrement.
- bool operator== (const const_iterator &other) const

  Equality comparison.
- bool operator!= (const const_iterator &other) const

  Inequality comparison.

Friends

- class **skip_list**

2.1.1 Detailed Description

template<typename T, typename Compare = std::less<T>, typename Allocator = std::allocator<T>>
class skip_list< T, Compare, Allocator >::const_iterator

Constant bidirectional iterator for skip_list.

The documentation for this class was generated from the following file:

- SkipList.h

2.2 skip_list< T, Compare, Allocator >::iterator Class Reference

Bidirectional iterator for skip_list.

`#include <SkipList.h>`

Public Types

- using **iterator_category** = std::bidirectional_iterator_tag
- using **value_type** = T
- using **difference_type** = std::ptrdiff_t
- using **pointer** = T *
- using **reference** = T &

Public Member Functions

- **iterator** (skip_list_node *node)
- reference operator* () const

  Dereference operator.
- pointer operator-> () const

  Member access operator.
- iterator & operator++ ()

  Prefix increment.
- iterator operator++ (int)

  Postfix increment.
- iterator & operator-- ()

  Prefix decrement.
- iterator operator-- (int)

  Postfix decrement.
- bool operator== (const iterator &other) const

  Equality comparison.
- bool operator!= (const iterator &other) const

  Inequality comparison.

Friends

- class **skip_list**

2.2.1 Detailed Description

template<typename T, typename Compare = std::less<T>, typename Allocator = std::allocator<T>>
class skip_list< T, Compare, Allocator >::iterator

Bidirectional iterator for skip_list.

The documentation for this class was generated from the following file:

- SkipList.h

2.3 skip_list< T, Compare, Allocator > Class Template Reference

A skip list container that provides logarithmic time complexity for search operations.

`#include <SkipList.h>`

Classes

- class const_iterator
  Constant bidirectional iterator for skip_list.
- class iterator
  Bidirectional iterator for skip_list.

Public Member Functions

- skip_list (const Compare &comp=Compare(), const Allocator &alloc=Allocator())
  Default constructor.
- skip_list (const Allocator &alloc)
  Constructor with allocator.
- skip_list (const skip_list &other)
  Copy constructor.
- skip_list (skip_list &&other) noexcept
  Move constructor.
- ~skip_list ()
  Destructor.
- skip_list & operator= (const skip_list &other)
  Copy assignment operator.
- skip_list & operator= (skip_list &&other) noexcept
  Move assignment operator.
- Allocator get_allocator () const noexcept
  Returns the container's allocator.
- bool empty () const noexcept

Checks if container is empty.

- std::size_t size () const noexcept

  Returns number of elements.

- iterator insert (const T &value)

  Inserts an element.

- iterator insert (T &&value)

  Inserts an element (move version)

- void clear ()

  Clears all elements from container.

- iterator erase (iterator pos)

  Erases element at position.

- void swap (skip_list &other) noexcept

  Swaps contents with another skip_list.

- iterator find (const T &value)

  Finds element with specific value.

- const_iterator find (const T &value) const

  Finds element with specific value (const version)

- bool contains (const T &value) const

  Checks if an element exists in the container.

- template<typename... Args>
  iterator emplace (Args &&... args)

  Inserts an element using in-place construction.

- iterator push_front (const T &value)

  Inserts an element at the beginning of the container.

- iterator push_back (const T &value)

  Inserts an element at the end of the container.

- void pop_front ()

  Removes the first element from the container.

- void pop_back ()

  Removes the last element from the container.

- void resize (std::size_t count, const T &value=T())

  Resizes the container to contain count elements.

- iterator begin () noexcept

  Returns iterator to beginning.

- const_iterator begin () const noexcept

  Returns const iterator to beginning.

- const_iterator cbegin () const noexcept

  Returns const iterator to beginning.

- iterator end () noexcept

  Returns iterator to end.

- const_iterator end () const noexcept

  Returns const iterator to end.

- const_iterator cend () const noexcept

  Returns const iterator to end.

- bool operator== (const skip_list &other) const

  Equality operator.

- bool operator!= (const skip_list &other) const

  Inequality operator.

2.3.1 Detailed Description

template<typename T, typename Compare = std::less<T>, typename Allocator = std::allocator<T>>
class skip_list< T, Compare, Allocator >

A skip list container that provides logarithmic time complexity for search operations.

Template Parameters

T	The type of elements stored in the container.
Compare	Comparison function object type (default: std::less<T>).
Allocator	Allocator type (default: std::allocator<T>).

2.3.2 Member Function Documentation

2.3.2.1 contains()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
bool skip_list< T, Compare, Allocator >::contains (
            const T & value ) const  [inline]
```

Checks if an element exists in the container.

Parameters

value	Value to search for

Returns

true if element exists, false otherwise

2.3.2.2 emplace()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
template<typename...  Args>
iterator skip_list< T, Compare, Allocator >::emplace (
            Args &&...  args )  [inline]
```

Inserts an element using in-place construction.

Template Parameters

Args	Argument types for element construction

Parameters

args	Arguments to forward to element constructor

Returns

Iterator to the inserted element

2.3.2.3 insert() [1/2]

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
iterator skip_list< T, Compare, Allocator >::insert (
            const T & value )  [inline]
```

Inserts an element.

Parameters

value	Value to insert

Returns

Iterator to the inserted element

2.3.2.4 insert() [2/2]

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
iterator skip_list< T, Compare, Allocator >::insert (
            T && value )  [inline]
```

Inserts an element (move version)

Parameters

value	Value to insert

Returns

Iterator to the inserted element

2.3.2.5 push_back()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
iterator skip_list< T, Compare, Allocator >::push_back (
            const T & value )  [inline]
```

Inserts an element at the end of the container.

Parameters

value	Value to insert

Returns

Iterator to the inserted element

2.3.2.6 push_front()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
iterator skip_list< T, Compare, Allocator >::push_front (
            const T & value )  [inline]
```

Inserts an element at the beginning of the container.

Parameters

value	Value to insert

Returns

Iterator to the inserted element

2.3.2.7 resize()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
```

```
void skip_list< T, Compare, Allocator >::resize (
            std::size_t count,
            const T & value = T() )  [inline]
```

Resizes the container to contain count elements.

Parameters

count	New size of the container
value	Value to initialize new elements with

2.3.2.8 swap()

```
template<typename T , typename Compare = std::less<T>, typename Allocator = std::allocator<↩
T>>
void skip_list< T, Compare, Allocator >::swap (
            skip_list< T, Compare, Allocator > & other )  [inline], [noexcept]
```

Swaps contents with another skip_list.

Parameters

other	Another skip_list to swap with

The documentation for this class was generated from the following file:

- SkipList.h

Index

contains
skip_list< T, Compare, Allocator >, 7

emplace
skip_list< T, Compare, Allocator >, 7

insert
skip_list< T, Compare, Allocator >, 8

push_back
skip_list< T, Compare, Allocator >, 9
push_front
skip_list< T, Compare, Allocator >, 9

resize
skip_list< T, Compare, Allocator >, 9

skip_list< T, Compare, Allocator >, 5
contains, 7
emplace, 7
insert, 8
push_back, 9
push_front, 9
resize, 9
swap, 10
skip_list< T, Compare, Allocator >::const_iterator, 3
skip_list< T, Compare, Allocator >::iterator, 4
swap
skip_list< T, Compare, Allocator >, 10