G++ 2.91.57，cygnus\cygwin-b20\include\g++**algo.h** 完整列表

```
/*
 *
 * Copyright (c) 1994
 * Hewlett-Packard Company
 *
 * Permission to use, copy, modify, distribute and sell this software
 * and its documentation for any purpose is hereby granted without fee,
 * provided that the above copyright notice appear in all copies and
 * that both that copyright notice and this permission notice appear
 * in supporting documentation.  Hewlett-Packard Company makes no
 * representations about the suitability of this software for any
 * purpose.  It is provided "as is" without express or implied warranty.
 *
 *
 * Copyright (c) 1996,1997
 * Silicon Graphics Computer Systems, Inc.
 *
 * Permission to use, copy, modify, distribute and sell this software
 * and its documentation for any purpose is hereby granted without fee,
 * provided that the above copyright notice appear in all copies and
 * that both that copyright notice and this permission notice appear
 * in supporting documentation.  Silicon Graphics makes no
 * representations about the suitability of this software for any
 * purpose.  It is provided "as is" without express or implied warranty.
 */

#ifndef __SGI_STL_ALGO_H
#define __SGI_STL_ALGO_H

#include <algobase.h>
#include <tempbuf.h>
#include <stl_algo.h>
#include <stl_numeric.h>

#ifdef __STL_USE_NAMESPACES

// Names from <stl_algo.h>
using __STD::for_each;
using __STD::find;
using __STD::find_if;
using __STD::adjacent_find;
using __STD::count;
using __STD::count_if;
using __STD::search;
using __STD::search_n;
using __STD::swap_ranges;
using __STD::transform;
```

```
using __STD::replace;
using __STD::replace_if;
using __STD::replace_copy;
using __STD::replace_copy_if;
using __STD::generate;
using __STD::generate_n;
using __STD::remove;
using __STD::remove_if;
using __STD::remove_copy;
using __STD::remove_copy_if;
using __STD::unique;
using __STD::unique_copy;
using __STD::reverse;
using __STD::reverse_copy;
using __STD::rotate;
using __STD::rotate_copy;
using __STD::random_shuffle;
using __STD::random_sample;
using __STD::random_sample_n;
using __STD::partition;
using __STD::stable_partition;
using __STD::sort;
using __STD::stable_sort;
using __STD::partial_sort;
using __STD::partial_sort_copy;
using __STD::nth_element;
using __STD::lower_bound;
using __STD::upper_bound;
using __STD::equal_range;
using __STD::binary_search;
using __STD::merge;
using __STD::inplace_merge;
using __STD::includes;
using __STD::set_union;
using __STD::set_intersection;
using __STD::set_difference;
using __STD::set_symmetric_difference;
using __STD::min_element;
using __STD::max_element;
using __STD::next_permutation;
using __STD::prev_permutation;
using __STD::find_first_of;
using __STD::find_end;
using __STD::is_sorted;
using __STD::is_heap;

// Names from stl_heap.h
using __STD::push_heap;
using __STD::pop_heap;
```

```
using __STD::make_heap;
using __STD::sort_heap;

// Names from <stl_numeric.h>
using __STD::accumulate;
using __STD::inner_product;
using __STD::partial_sum;
using __STD::adjacent_difference;
using __STD::power;
using __STD::iota;

#endif /* __STL_USE_NAMESPACES */

#endif /* __SGI_STL_ALGO_H */

// Local Variables:
// mode:C++
// End:
```