

标准库名字和头文件_算法简介_再谈IO库计算机等级考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E6_A0_87_E5_87_86_E5_BA_93_E5_c97_645565.htm 本文中用到的标准库

名字及其头文件：
: abort, : accumulate, inner_product, : allocator, auto_ptr, uninitialized_copy, : back_inserter, front_inserter, inserter, istream_iterator, ostream_iterator, reverse_iterator, : bad_alloc, : bad_cast, type_info, : bind2nd, less_equal, negate, not1, plus, : bitset, : boolalpha, cerr, cin, cout, dec, endl, ends, fixed, flush, hex, internal, istream, left, noboolalpha, noshowbase, noskipws, nunitbuf, nouppercase, oct, ostream, right, scientific, showbase, showpoint, skipws, unitbuf, uppercase : copy, count, count_if, equal_range, fill, fill_n, find, find_end, find_first_of, for_each, max, main, nth_element, partial_sort, replace, replace_copy, set_difference, set_intersection, set_union, sort, stable_sort, unique, unique_copy, upper_bound, : deque, : exception, unexpected, : fstream, ifstream, ofstream, : getline, string, : ios_base, : isalpha, islower, ispunct, isspace, isupper, : istringstream, ostringstream, stringstream, : list, : logic_error, out_of_range, range_error, runtime_error, : make_pair, pair, : map, multimap : multiset, set : priority_queue, queue : ptrdiff_t, size_t, : setfill, setprecision, setw, : sqrt, : stack, : strcmp, strcpy, strlen, strncpy, : vector
附录 A.2 算法简介 标准库定义了 100 多个算法，要学习如何使用它们，需要理解它们的结构，而不是记住每个算法的细节（1）查找对象的算法：
find(begin,end,val)
count(begin,end,val) find_if(begin,end,unaryPred)

count_if(beg,end,unaryPred) find_first_of(beg1,end1,beg2,end2)
find_first_of(beg1,end1,beg2,end2,binaryPred)
find_end(beg1,end1,beg2,end2)
find_end(beg1,end1,beg2,end2,binaryPred) adjacent_find(beg,end)
adjacent_find(beg,end,binaryPred) search(beg1,end1,beg2,end2)
search(beg1,end1,beg2,end2,binaryPred)
search_n(beg,end,count,val)
search_n(beg,end,count,val,binaryPred) (2) 其他只读算法 :
for_each(beg,end,f) mismatch(beg1,end1,beg2)
mismatch(beg1,end1,beg2,binaryPred) equal(beg1,end1,beg2)
equal(beg1,end1,beg2,binaryPred) (3) 二分查找算法 :
lower_bound(beg,end,val) lower_bound(beg,end,val,comp)
upper_bound(beg,end,val) upper_bound(beg,end,val,comp)
equal_range(beg,end,val) equal_range(beg,end,val,comp)
binary_search(beg,end,val) binary_search(beg,end,val,comp) (4)
写容器元素的算法 : fill_n(dest,cnt,val)
generate_n(dest,cnt,Gen) copy(beg,end,dest)
transform(beg,end,dest,unaryOp)
transform(beg,end,beg2,dest,binaryOp)
replace_copy(beg,end,dest,old_val,new_val)
replace_copy_if(beg,end,dest,unaryPred,new_val)
merge(beg1,end1,beg2,end2,dest)
merge(beg1,end1,beg2,end2,dest,comp) swap(elem1,elem2)
iter_swap(iter1,iter2) swap_ranges(beg1,end1,beg2) fill(beg,end,val)
generate(beg,end,Gen) replace(beg,end,old_val,new_val)
replace_if(beg,end,unaryPred,new_val)

copy_backward(beg,end,dest) inplace_merge(beg,mid,end)
inplace_merge(beg,mid,end,comp) (5) 划分与排序算法 :
stable_partition(beg,end,unaryPred) partition(beg,end,unaryPred)
sort(beg,end) stable_sort(beg,end) sort(beg,end,comp)
stable_sort(beg,end,comp) partial_sort(beg,mid,end)
partial_sort(beg,mid,end,comp)
partial_sort_copy(beg,end,destBeg,destEnd)
partial_sort_copy(beg,end,destBeg,destEnd,comp)
nth_element(beg,nth,end) nth_element(beg,nth,end,comp) (6)
通用重新排序算法 : remove(beg,end,val)
remove_if(beg,end,unaryPred) unique(beg,end)
unique(beg,end,binaryPred) rotate(beg,mid,end) reverse(beg,end)
reverse_copy(beg,end,dest) remove_copy(beg,end,dest,val)
remove_copy_if(beg,end,dest,unaryPred)
unique_copy(beg,end,dest) unique_copy(beg,end,dest,binaryPred)
rotate_copy(beg,mid,end,dest) random_shuffle(beg,end)
random_shuffle(beg,end,rand) (7) 排列算法 :
next_permutation(beg,end) next_permutation(beg,end,comp)
prev_permutation(beg,end) prev_permutation(beg,end,comp) (8)
有序序列的集合算法 : includes(beg,end,beg2,end2)
includes(beg,end,beg2,end2,comp)
set_union(beg,end,beg2,end2,dest)
set_union(beg,end,beg2,end2,dest,comp)
set_intersection(beg,end,beg2,end2,dest)
set_intersection(beg,end,beg2,end2,dest,comp)
set_difference(beg,end,beg2,end2,dest)

set_difference(beg,end,beg2,end2,dest,comp)

set_symmetric_difference(beg,end,beg2,end2,dest)

set_symmetric_difference(beg,end,beg2,end2,dest,comp) (8) 最大值和最小值算法 : min(va1,va2) min(val1,val2,comp)

max(val1,val2) max(val1,val2,comp) min_element(beg,end)

min_element(beg,end,comp) max_element(beg,end)

max_element(beg,end,comp)

lexicographical_compare(beg1,end1,beg2,end2)

lexicographical_compare(beg1,end1,beg2,end2,comp) (10) 算术算法 : 在中 accumulate(beg,end,init)

accumulate(beg,end,init,BinaryOp)

inner_product(beg1,end1,beg2,init)

inner_product(beg1,end1,beg2,init,BinOp1,BinOp2)

partial_sum(beg,end,dest) partial_sum(beg,end,dest,BinaryOp)

adjacent_difference(beg,end,dest)

adjacent_difference(beg,end,dest,BinaryOp) 总结 : 查找对象的

算法、其他只读算法、二分查找算法、写容器元素的算法、

划分与排序算法、通用重新排序算法、排列算法、有序序列

的集合算法、最大值和最小值算法、算术算法 附录 A.3 再谈

IO 库 iostream中定义的操纵符 :

boolalpha,noboolalpha,showbase,noshowbase,showpoint,noshowp

oint,showpos,noshowpos,uppercase,nouppercase,dec,hex,oct,left,ri

ght,internal,fixed,scientific,flush,ends,endl,unitbuf,nounitbuf,skipws,

noskipws,ws iomanip中定义的操纵符 :

setfill(ch),setprecision(n),setw(w),setbase(b) 改变流格式状态的操

纵符通常为后续 IO保留改变后的格式状态 取消操纵符的任何

状态改变通常是最好的。一般而言，流应该在每个 IO 操作之后处于通常的默认状态 用 flags 操作恢复格式状态：flags() 返回流的当前格式状态，返回值是名为 fmtflags 的标准库类型。flags(arg) 将流格式置为指定格式 控制输出格式：有两大类，控制数值的表示，控制填充符的数量和布局 单字节低级 IO 操作：

is.get(ch), os.put(ch), is.get(), is.putback(ch), is.unget(), is.peek() 文件结束符：EOF，在中定义 多字节操作：

is.get(sink, size, delim), is.getline(sink, size, delim), is.read(sink, size), is.gcount(), os.write(source, size), is.ignore(size, delim) 流的随机访问：函数 seekg, tellg, seekp, tellp。普通 istream 对象一般不允许随机访问 在大多数系统上，绑定到 cin, cout, cerr 和 clog 的流不支持随机访问。随机访问特性常用于 fstream 和 sstream。编辑特别推荐: 送给正在学习 C 朋友的 50 条建议 C 笔试考前练习 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com