4.链表(list)及其成员函数

1)唯一化

void unique(void);

将连续重复出现的元素做唯一化。

1. 排序

void sort(void); // 通过"<"比大小

void sort(LESS less); // 通过比较器less比大小

3)拆分

将参数列表中的部分或全部元素剪切到调用列表pos位置前。

void splice(IT pos, list& lst);

void splice(IT pos, list& lst, IT del);

void splice(IT pos, list& lst, IT begin, IT end);

时间复杂度：O(1)(因为是链表，仅交换几个指针)

4)合并

将有序参数列表中的全部元素剪切到有序调用列表中，保持后者有序。

void merge(list& lst);

void merge(list& lst, LESS less);

时间复杂度：O(N)

--------------------------

5.堆栈(stack)

#include <stack>

stack<元素类型，[底层容器类型]> 堆栈对象(构造实参表);

底层容器：vector/deque(默认)/list

注解：底层容器就是底层的存储结构，选什么不影响逻辑结果。

push -> push\_back

pop -> pop\_back

top -> back

size -> size

empty -> empty

clear -> clear

参见：stack.cpp

--------------------------

6.队列(queue)

#include <queue>

queue<元素类型，[底层容器类型]> 队列对象(构造实参表);

底层容器：deque(默认)/list

push -> push\_back

pop -> pop\_front

back -> back

front -> front

size -> size

empty -> empty

clear -> clear

参见：queue.cpp

7.优先队列

#include <queue>

priority\_queue<元素类型，[底层容器类型], 比较器类型]> 优先队列对象 (构造实参表);

底层容器：vector/deque(默认)

优者先出，默认以大者为优，也可以通过比较器定制优先级(内部使用小于号比较)。

原理：入队时已经使用了插入排序算法。

参见：pqueue.cpp

------------------------

8.映射(map)

1)逻辑模型：一一对应，键(信息索引)值(信息内容)对，用于信息检索，检索性能可以达到对数级(O(logN))。

2)物理模型：平衡有序二叉树，又名红黑树。

平衡有序二叉树：任何一个节点上的左子树和右子树中，元素个数最多相差一个。

3

2 4

1 5

3)键必须是唯一的。

4)迭代过程实际上是关于键的中序遍历(L-D-R)，键升序。

5)存储单位是由键和值组成的pair对象。

template<typename FIRST, typename SECOND>

class pair{

public:

pair(FIRST const& f, SECOND const& s):first(f),second(s){}

FIRST first; // 键

SECOND second; // 值

};

映射的迭代器相当于是指向pair对象的指针。

6)映射中键都是只读的。

7)构建和修改性能较差。适用于结构稳定但是频繁检索的场合。

8)支持"下标"运算，用键做下标，得到对应的值的引用。如果所给出键不存在，会增加一个节点，返回其值，如果键存在，直接返回对应的值。

9）mapped\_type::key\_type/mapped\_type::value\_type（返回key类型和value类型）

参见：map.cpp

------------------------

9.多重映射(multimap)

允许键重复的映射，表示一对多的逻辑关系。不支持下标运算符。

根据键查找匹配的元素集合：

pair<IT, IT> equal\_range(KEY const& key);

返回值：一个pair对象，内含两个迭代器(前者first起始迭代器和后者second终止迭代器)。

first指向第一个和参数key匹配的元素；

second指向最后一个和参数key匹配的元素的下一个元素。

IT lower\_bound (KEY const& key); // 匹配下限迭代器(begin)(了解)

IT upper\_bound (KEY const& key); // 匹配上限迭代器(end)

参见：mmap.cpp

------------------------

10.集合(set)

没有值只有键的映射。

集合会自动排重和排序(平衡有序二叉树)，向集合中插入重复元素会被自动过滤。

其迭代器返回常引用。

11.多重集合(multiset)

没有值只有键的多重映射。

多重集合也会自动排序。

参见：set.cpp