程序设计实习

郭炜　微博 http://weibo.com/guoweiofpku
http://blog.sina.com.cn/u/3266490431
刘家瑛 微博 http://weibo.com/pkuliujiaying

标准模板库STL

概述

泛型程序设计

C++ 语言的核心优势之一就是便于软件的重用

C++中有两个方面体现重用：

1. 面向对象的思想：继承和多态，标准类库

2. 泛型程序设计(generic programming) 的思想： 模板机制，以及标准模板库 STL

泛型程序设计

简单地说就是使用模板的程序设计法。

将一些常用的数据结构（比如链表，数组，二叉树）和算法（比如排序，查找）写成模板，以后则不论数据结构里放的是什么对象，算法针对什么样的对象，则都不必重新实现数据结构，重新编写算法。

标准模板库（Standard Template Library）就是一些常用数据结构和算法的模板的集合。

有了STL，不必再写大多的标准数据结构和算法，并且可获得非常高的性能。

STL中的基本的概念

容器：可容纳各种数据类型的通用数据结构，是类模板　Container

迭代器：可用于依次存取容器中元素，类似于指针　iterator

算法：用来操作容器中的元素的函数模板

- sort()来对一个vector中的数据进行排序
- find()来搜索一个list中的对象

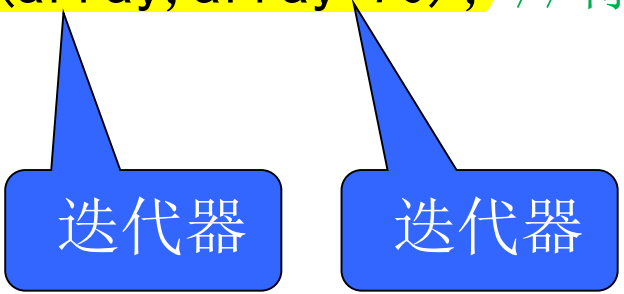
算法本身与他们操作的数据的类型无关，因此他们可以在从简单数组到高度复杂容器的任何数据结构上使用。

STL中的基本的概念

```
int array[100];
```

该数组就是容器，而 int * 类型的指针变量就可以作为迭代器，sort 算法可以作用于该容器上，对其进行排序：

```
sort(array,array+70); //将前70个元素排序
```



容器概述

可以用于存放各种类型的数据（基本类型的变量，对象等）的数据结构，都是类模版，分为三种：

1）顺序容器

vector，deque，list

2）关联容器

set，multiset，map，multimap

3）容器适配器

stack，queue，priority_queue

容器概述

对象被插入容器中时，被插入的是对象的一个复制品。许多算法，比如排序，查找，要求对容器中的元素进行比较，有的容器本身就是排序的，所以，放入容器的对象所属的类，往往还应该重载 == 和 < 运算符。

顺序容器简介

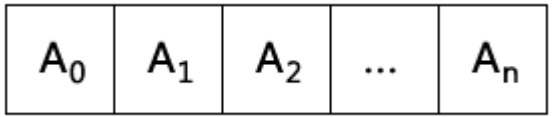
容器并非排序的，元素的插入位置同元素的值无关。

有vector,deque,list 三种

- vector 头文件〈vector〉

size is changeable

动态数组。元素在内存连续存放。随机存取任何元素都能在常数时间完成。在尾端增删元素具有较佳的性能(大部分情况下是常数时间）。



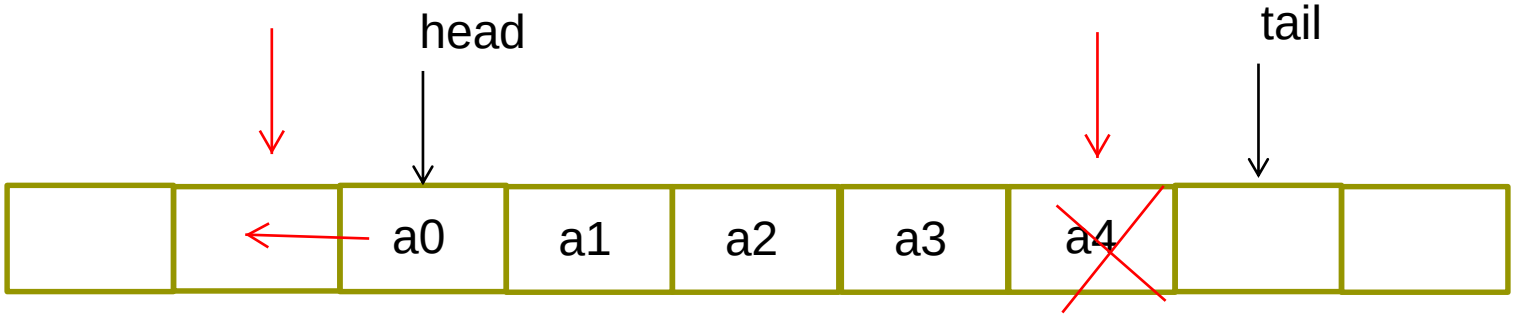
O(1)

顺序容器简介

slower than vector

- deque　头文件 <deque>

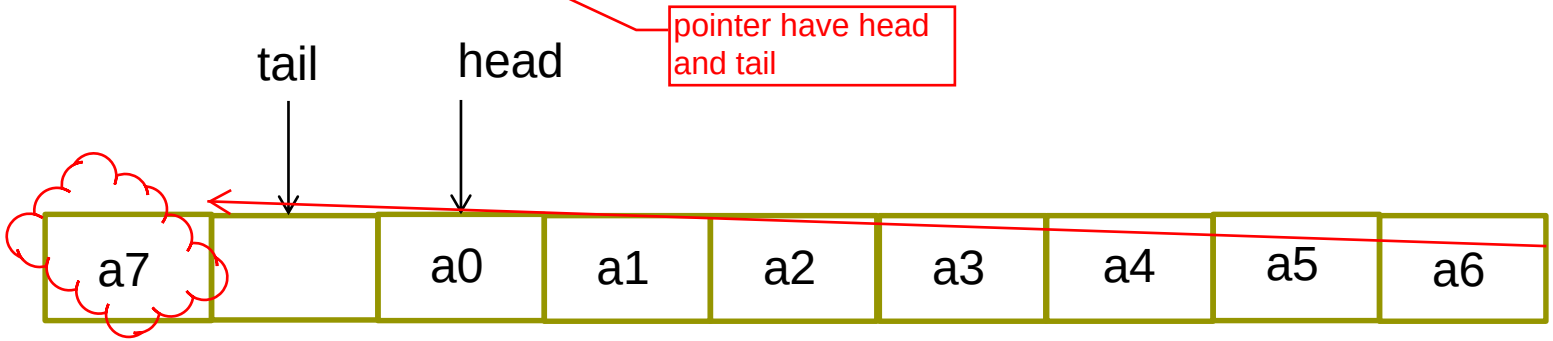
双向队列。元素在内存连续存放。随机存取任何元素都能在常数时间完成(但次于vector)。在两端增删元素具有较佳的性能(大部分情况下是常数时间)。



顺序容器简介

- deque　　头文件 <deque>

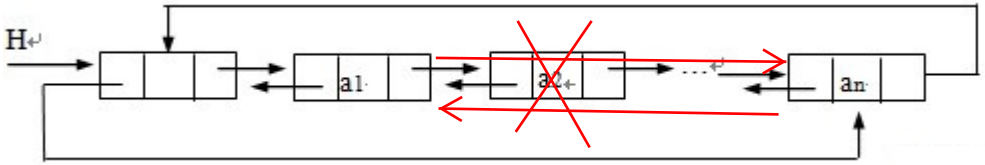
双向队列。元素在内存连续存放。随机存取任何元素都能在常数时间完成(但次于vector)。在两端增删元素具有较佳的性能(大部分情况下是常数时间)。



顺序容器简介

- list　　头文件 <list>

双向链表。元素在内存不连续存放。在任何位置增删元素都能在常数时间完成。不支持随机存取。



关联容器简介

- 元素是排序的
- 插入任何元素，都按相应的排序规则来确定其位置
- 在查找时具有非常好的性能
- 通常以平衡二叉树方式实现，插入和检索的时间都是 O(log(N))

- set/multiset　头文件 <set>

  set 即集合。set中不允许相同元素，multiset中允许存在相同的元素。

- map/multimap　头文件 <map>

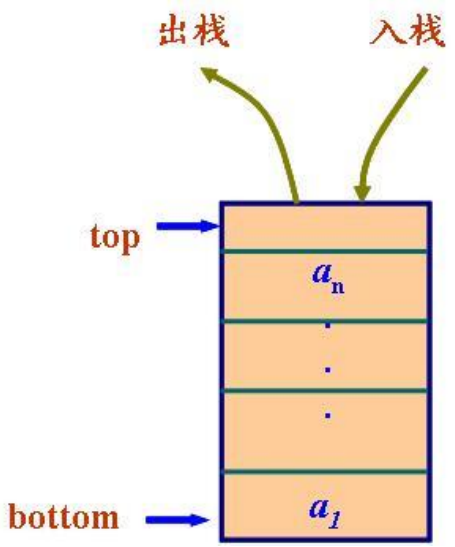
  map与set的不同在于map中存放的元素有且仅有两个成员变量，一个名为first，另一个名为second，map根据first值对元素进行从小到大排序，并可快速地根据first来检索元素。

  map同multimap的不同在于是否允许相同first值的元素。

容器适配器简介

- stack ：头文件 <stack>

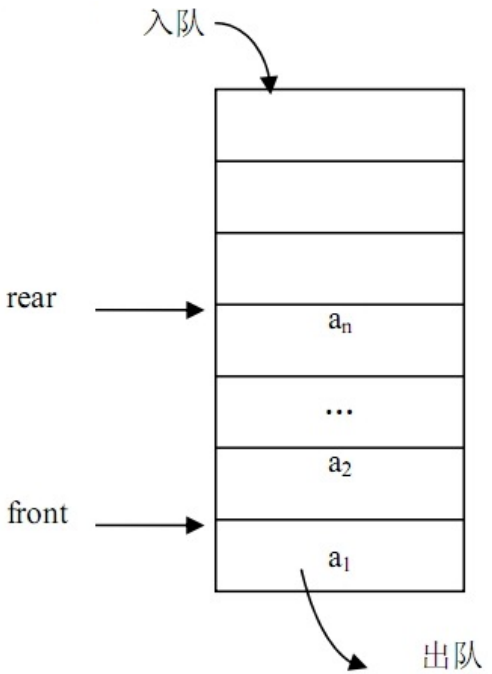
  栈。是项的有限序列，并满足序列中被删除、检索和修改的项只能是最近插入序列的项（栈顶的项）。后进先出。



容器适配器简介

- queue 头文件〈queue〉

  队列。插入只可以在尾部进行，删除、检索和修改只允许从头部进行。先进先出。



容器适配器简介

- priority_queue 头文件 <queue>

  优先级队列。最高优先级元素总是第一个出列

顺序容器和关联容器中都有的成员函数

begin	返回指向容器中第一个元素的迭代器
end	返回指向容器中最后一个元素后面的位置的迭代器
rbegin	返回指向容器中最后一个元素的迭代器
rend	返回指向容器中第一个元素前面的位置的迭代器
erase	从容器中删除一个或几个元素
clear	从容器中删除所有元素

顺序容器的常用成员函数

front ：返回容器中第一个元素的引用

back ： 返回容器中最后一个元素的引用

push_back ： 在容器末尾增加新元素

pop_back ： 删除容器末尾的元素

erase ：删除迭代器指向的元素(可能会使该迭代器失效)，或删除一个区间，返回被删除元素后面的那个元素的迭代器

迭代器

- 用于指向顺序容器和关联容器中的元素
- 迭代器用法和指针类似
- 有const 和非 const两种
- 通过迭代器可以读取它指向的元素
- 通过非const迭代器还能修改其指向的元素

迭代器

定义一个容器类的迭代器的方法可以是：

```
容器类名::iterator  变量名;
```

或：

```
容器类名::const_iterator 变量名;
```

访问一个迭代器指向的元素：

```
* 迭代器变量名
```

迭代器

迭代器上可以执行 ++ 操作，以使其指向容器中的下一个元素。
如果迭代器到达了容器中的最后一个元素的后面，此时再使用它，就会出错，类似于使用NULL或未初始化的指针一样。

迭代器示例

```
#include <vector>
#include <iostream>
using namespace std;

int main()  {
    vector<int> v; //一个存放int元素的数组，一开始里面没有元素
    v.push_back(1);  v.push_back(2);   v.push_back(3);  v.push_back(4);
    vector<int>::const_iterator i;  //常量迭代器
    for( i = v.begin();i != v.end();++i )
        cout << * i << ",";
    cout << endl;
```

输出结果：

1,2,3,4,

4,3,2,1,

100,100,100,100,

```
    vector<int>::reverse_iterator r;  //反向迭代器
    for( r = v.rbegin();r != v.rend();r++ )
        cout << * r << ",";
    cout << endl;
    vector<int>::iterator j;  //非常量迭代器
    for( j = v.begin();j != v.end();j ++ )
        * j =  100;
    for( i = v.begin();i != v.end();i++ )
        cout << * i << ",";
}
```

输出结果：

1,2,3,4,

4,3,2,1,

100,100,100,100,

双向迭代器

若p和p1都是双向迭代器，则可对p、p1可进行以下操作：

++p，p++	使p指向容器中下一个元素
--p，p--	使p指向容器中上一个元素
* p	取p指向的元素
p = p1	赋值
p == p1 ，p!= p1	判断是否相等、不等

随机访问迭代器

若p和p1都是随机访问迭代器，则可对p、p1可进行以下操作：

- 双向迭代器的所有操作
- p += i 将p向后移动i个元素
- p -= i 将p向向前移动i个元素
- p + i 值为：指向 p 后面的第i个元素的迭代器
- p - i 值为：指向 p 前面的第i个元素的迭代器
- p[i] 值为：p后面的第i个元素的引用
- p < p1，p <= p1，p > p1，p>= p1

In-Video Quiz

1. map、set、list、vector、deque这五种容器中，有几种是排序的？
A) 1 B) 2 C) 3 D) 4

2. 下面5种操作，有几种时间复杂度一定是常数的？
(1)在vector尾部增删元素
(2)在关联容器中查找元素
(3)在list中的某一位置增删元素
(4)随机访问vector中的元素
(5)在multimap中插入元素
A) 1 B) 2 C) 3 D)4