STL的注意事项!!!!

1.vector

<1> vector的insert()和erase()是 $O(\sqrt{n})$;

2.map

<1> 确定某位有值时不能用mp[x]而是要用mp.count(x) (会出现map里面装了零，你却当它不存在)

3.set

<1> 当在set中使用lower_bound什么的时候

```
lower_bound(S.begin(),S.end(),v);
S.lower_bound(v);
```

第一个时间复杂度$O(\log^2 n)$

第二个时间复杂度$O(\log n)$

tip:原因是你把set当做数组用了查询$O(\log n)$

<2> 在set中重载小于号的时候记得把两个关键字都写上，否则会有奇怪的去重现象。

```
bool operator < (struct x,struct y){
    return x.first<y.first
}//就算第二关键字不一样SET也会当做一样的去重
bool operator < (struct x,struct y){
    return (x.first==y.first)?x.second<y.second:x.first<y.first;
}//所以就算二关键字不重要也请打成这样子
```

tip:另附不要想着去重载什么"=="SET不认它们

4.multiset

<1>如果要删除单个元素的单次而不是一整个元素。

```
S.erase(S.find(val));//仅删除一个
S.erase(val);//整个元素删除
```

c++参考手册

GFOJ-STL总结

向量(vector) 连续存储的元素

列表(list) 由节点组成的双向链表，每个结点包含着一个元素

双端队列(deque) 连续存储的指向不同元素的指针所组成的数组

适配器容器

栈(stack) 后进先出的值的排列

队列(queue) 先进先出的值的排列

优先队列(priority_queue) 元素的次序是由作用于所存储的值对上的某种谓词决定的的一种队列

关联式容器

集合(set) 由节点组成的红黑树，每个节点都包含着一个元素，节点之间以某种作用于元素对的谓词排列，没有两个不同的元素能够拥有相同的次序

多重集合(multiset) 允许存在两个次序相等的元素的集合

映射(map) 由{键，值}对组成的集合，以某种作用于键对上的谓词排列

多重映射(multimap) 允许键对有相等的次序的映射