## Vector

1. 头文件未正确包含
2. istream、cin、ostream前要用std命名空间
3. 未找到匹配令牌 ：括号不匹配
4. 编译器内部错误：istream\_iterator(std::istream& s) : stream(&s) { read(); };在函数的实现里面read后面忘记加；
5. is\_integer<>使用模板特化来实现，一般情况下默认为false\_type，把为true\_type的类型全部特化
6. vector的at函数调用operator[]的方式为\*this[n]，不是operator[n]
7. C1903 无法从以前的错误中恢复；正在停止编译 错误太多了
8. try块中声明的变量在catch块中失效

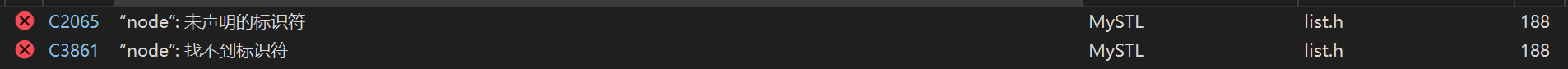
## list

1. reverse\_iterator的typedef要在const\_reverse\_iterator之后，否则会覆盖掉reverse\_iterator模板
2. 返回值是<> + ::时要用typedef编译器才知道返回值是一个变量

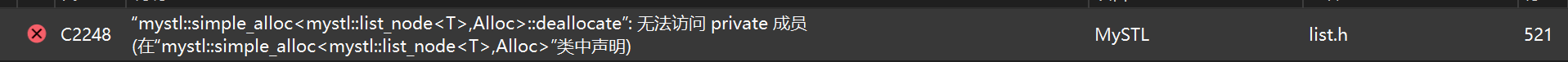




1. list\_iterator中继承的node不能直接使用，要用this->node，因为子类也是模板，父类只能访问具体一个类的成员，不能访问一个模板的成员。



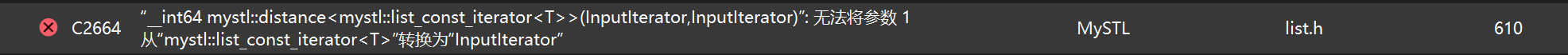
1. simple\_allocator的函数都要声明为public



1. Is\_integer<>中要填入具体的类型，不是具体的变量

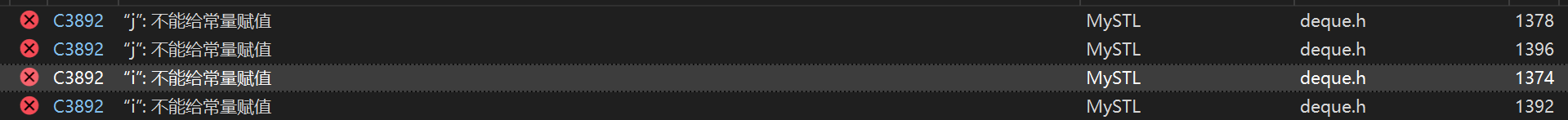


1. list\_const\_iterator的复制构造函数不正确(参数没有加const)



## deque

1. typedef时把size\_t前加了const



## Rb\_tree

1. 静态成员函数不允许使用修饰符（如const）。因为static成员不是任何对象的组成部分，将成员函数声明为const就是承诺不会修改该函数所属的对象。



1. root函数返回类型应该为base\_ptr，只有base\_ptr才能访问lchild和rchild。



1. for (; first != last; ++first) { erase(first); }

在删除操作中，迭代器 first 的值被 erase 函数改变了，因为 erase 函数会调整红黑树的结构，从而可能使迭代器失效。为了解决这个问题，可以在循环中提前保存下一个迭代器的值，然后再调用 erase 函数

## Map

1. 类模板的成员函数中没加分号等错误



## Hashtable

1. 类的声明中存在重定义默认参数的问题。在C++中，不允许在类的定义中重复指定默认参数。



1. 不能两个头文件相互包含。
2. 在hashtable的find函数中返回应该是const\_iterator(first, const\_cast<hashtable\*>(this));否则const\_iterator构造函数参数类型不匹配。

在hashtable\_const\_iterator的构造函数中要利用const\_cast将参数的const属性除去才能进行赋值。

其实归根结底是因为没有定义返回值为const\_iterator的find函数。

