

1 测试程序的设计思路

首先参考.h 文件中 ++ 算法的实现完成了对-的重载。关于.c 文件的书写。大致思路就是按照.h 文件中函数的功能依照顺序进行测试。首先我创建了一个空链表，调用 push 函数向其中插入了元素，检验插入函数是否可用。然后再调用 front 和 back 验证其输出是否正确。然后再调用 pop 删除掉对应元素再次检查。随后调用 insert 函数插入元素后再次打印检查。然后再检验拷贝构造函数，移动函数。然后检测了链表的初始化功能，通过打印首位元素来检验输出是否正确。然后分别对该链表进行清除操作和清空操作，来检测 erase 函数和 clear 函数的正确性。最后测试了迭代器对链表进行正向遍历和反向遍历，来检测 ++ 和-是否有正确重载。

2 测试的结果

测试结果一切正常。

输出为：

列表是否为空? true

push_front 和 push_back 后的大小: 3

前面的元素: 5

后面的元素: 15

pop_front 后, 前面的元素: 10

pop_back 后, 后面的元素: 10

插入 20 到前面后, 前面的元素: 20

复制后的列表前面的元素: 20

移动后的列表前面的元素: 20

移动后复制列表的大小: 0

赋值后的列表前面的元素: 20

初始化列表, 前面的元素: 30

初始化列表, 后面的元素: 50

删除第二个元素后, 前面的元素: 30

删除第二个元素后, 后面的元素: 50

清空列表后是否为空? true

rangeList 中的元素: 1 2 3 4 5

逆向顺序中的元素: 5 4 3 2 1

我用 valgrind 进行测试, 发现没有发生内存泄露。

3 bug 报告

一切正常。