

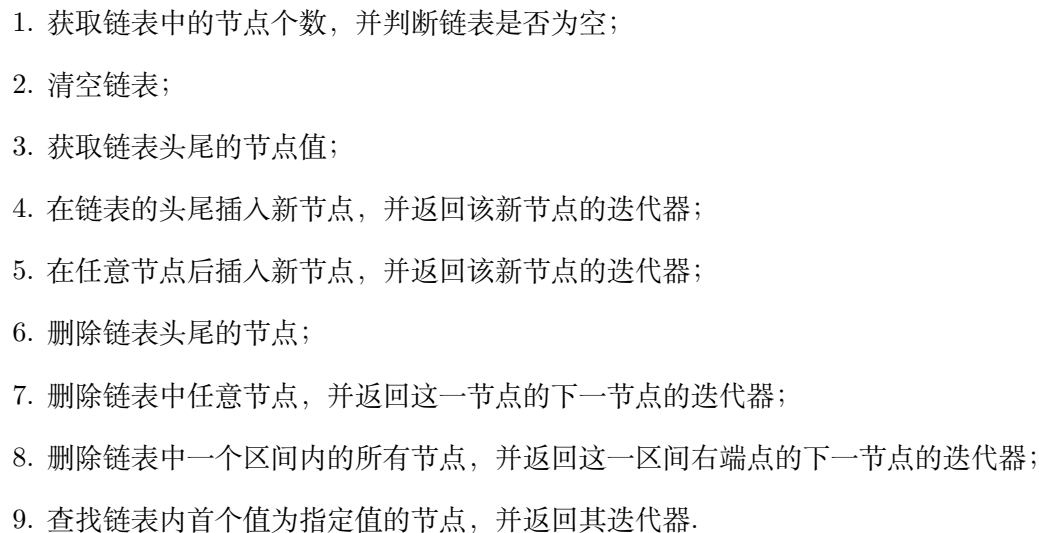
双链表的设计与实现实验报告

王笑同 数学与应用数学（强基计划）2101 3210105450

2022 年 10 月 5 日

1 实验内容

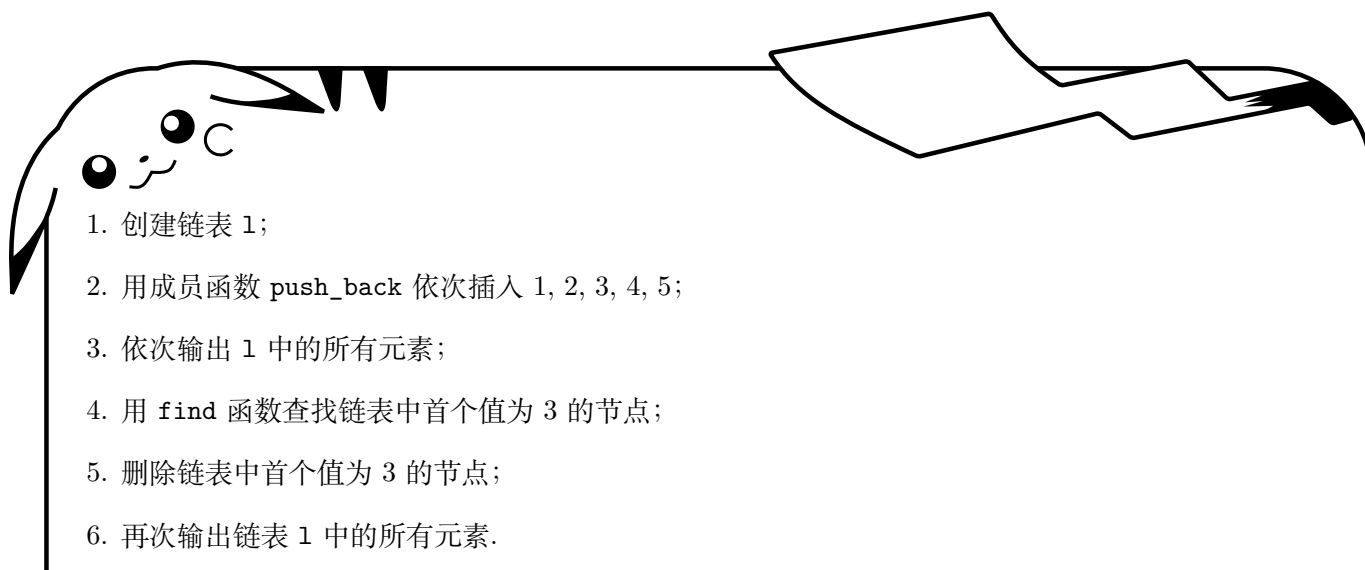
参考课本 91 页 3.5 节，在头文件 `DoubleLinkedList.h` 中实现了双链表的所有基本操作，并在测试程序 `main.cpp` 中实现了查找函数，这些操作包括：

- 
1. 获取链表中的节点个数，并判断链表是否为空；
 2. 清空链表；
 3. 获取链表头尾的节点值；
 4. 在链表的头尾插入新节点，并返回该新节点的迭代器；
 5. 在任意节点后插入新节点，并返回该新节点的迭代器；
 6. 删除链表头尾的节点；
 7. 删除链表中任意节点，并返回这一节点的下一节点的迭代器；
 8. 删除链表中一个区间内的所有节点，并返回这一区间右端点的下一节点的迭代器；
 9. 查找链表内首个值为指定值的节点，并返回其迭代器。

其中涉及到返回节点迭代器的操作，若对应节点不存在，则返回指向空节点的迭代器。

2 测试流程

为了测试头文件 `DoubleLinkedList.h`，我编写了测试代码 `main.cpp`，其中实现了以下操作：



在 Ubuntu 下得到输出结果如下图:

```
wxt@xtitxUbuntu:~/ds2022/list$ ./main
12345
1245
```

图 1: main.cpp 的运行结果

输出结果是符合预期的.

为了检测内存是否有泄露, 使用 valgrind 中的 memcheck 工具:

```
wxt@xtitxUbuntu:~/ds2022/list$ valgrind --tool=memcheck ./main
==20578== Memcheck, a memory error detector
==20578== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==20578== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==20578== Command: ./main
==20578==
12345
1245
==20578==
==20578== HEAP SUMMARY:
==20578==    in use at exit: 0 bytes in 0 blocks
==20578==   total heap usage: 9 allocs, 9 frees, 73,896 bytes allocated
==20578==
==20578== All heap blocks were freed -- no leaks are possible
==20578==
==20578== For lists of detected and suppressed errors, rerun with: -s
==20578== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

图 2: 检测内存是否泄露

结果输出 “No leaks are possible”，说明程序正常结束，无内存泄露可能。

3 注记 & 总结

为了增强代码的可读性，我为两份代码都添加了必要的英文注释，且都采用 doxygen 格式，让代码更规范。代码中也体现了一定的容错率，如 `find` 函数中，即使没有找到符合要求的节点，程序也不会发生任何意外错误。