

# 数据结构与算法第三次作业报告

Lingxi Gao \*

*Zhejiang University*

2024 年 10 月 11 日

## 问题描述

在目前的设计中，`remove()` 函数会删除 `currentPos` 之后的元素，而如果用户想先查看某个元素 `x` 是否在单链表中，如果在，则删除该元素，这一点目前的 `remove()` 函数无法做到。

## 改进思路

我的思路是重载一个 `remove(T _val)` 函数，实现以下功能：

输入 `_val` 的值，如果在链表中，则删除该值；如果不在链表中，则报错“the value you put is not in current list”并终止进程。

具体地，首先调用 `find()` 函数确认 `_val` 是否在当前 `SingleLinkedList` 中，由于执行完 `find()` 操作之后，如果 `_val` 在 `SingleLinkedList` 之中，`currentpos` 会指向 `_val` 所在的节点。之后再实现删除 `currentpos` 指向的节点即可。详细的代码如下：

Listing 1: 改进后的代码

```
template<typename T>
void SingleLinkedList<T>::remove(T _val){
    if (find(_val)){
        Node* p=head;
        if (head!=currentPos){
            while (p->next!=currentPos){
                p=p->next;
            }
            Node* p1=p->next->next;
            delete p->next;
            p->next=p1;
            currentPos=p;
            --size;
        }
    }
}
```

---

\*Electronic address: 3210105373@zju.edu.cn

```

        else{
            Node* p1=head->next;
            delete currentPos;
            head=p1;
            currentPos=head;
        }
    }
    else{
        cerr<<"the_value_you_put_is_not_in_current_single_linked_list"<<endl;
        exit(1);
    }
}

```

Listing 2: 测试

```

#include "LinkedList.h"
int main()
{
    SingleLinkedList<int> a1{1,2,3,4,5};
    a1.remove(3);
    a1.remove(1);
    a1.printList();
    return 0;
};

```

## 运行结果

最终运行结果如下:

2 4 5