

列表

无序列表：查找与去重

03-03

顧長康噉甘蔗，先食尾。問所以，云：“漸至佳境。”

有些事，你一辈子总也忘不掉。凡是让你揪心的事，在你身上，  
都会发生两次。或两次以上。

邓俊辉

deng@tsinghua.edu.cn

# 查找

❖ `template <typename T> //  $0 \leq n \leq \text{rank}(p) < \_size$`

`ListNodePosi<T> List<T>::find( T const & e, int n, ListNodePosi<T> p ) const {`

`while ( 0 < n-- ) //自后向前逐个比对`

`if ( e == ( p = p->pred ) ->data ) //假定类型T已重载 “==”`

`return p; //在p的n个前驱中，等于e的最靠后者`

`return NULL; //失败`

`} //O(n)`



❖ `ListNodePosi<T> find( T const & e ) const { return find( e, _size, trailer ); }`

# 去重

❖ template <typename T> int List<T>::deduplicate() {

int oldSize = \_size;

ListNodePosi<T> p = first();



for ( Rank r = 0; p != trailer; p = p->succ ) //O(n)

if ( ListNodePosi<T> q = find( p->data, r, p ) )

remove ( q );



else

r++; //无重前缀的长度



return oldSize - \_size; //删除元素总数

} //正确性及效率分析的方法与结论，与Vector::deduplicate()相同