

列表

无序列表：查找与去重

03-03

邓俊辉

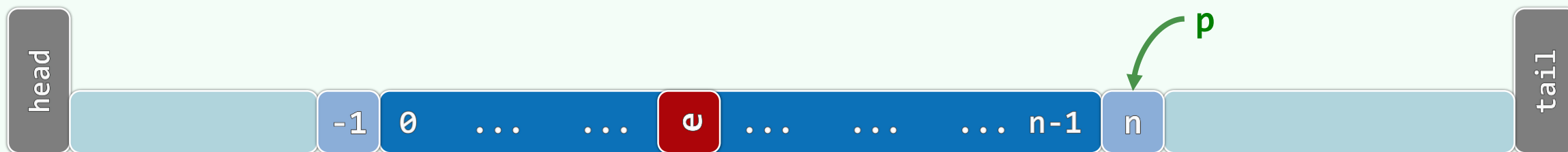
deng@tsinghua.edu.cn

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

# 查找

```
template <typename T> //0 <= n <= rank(p) < _size
ListNodePosi<T> List<T>::find( T const & e, Rank n, ListNodePosi<T> p ) const {
    while ( 0 < n-- ) //自后向前
        if ( e == ( p = p->pred ) ->data ) //逐个比对 (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)
```



```
template <typename T>
ListNodePosi<T> find( T const & e ) const { return find( e, _size, tail ); }
```

# 去重

```
template <typename T> Rank List<T>::dedup() {
```

```
    Rank oldSize = _size;
```

```
    ListNodePosi<T> p = first();
```



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

```
        if ( ListNodePosi<T> q = find( p->data, r, p ) ) //O(n)
```

```
            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::dedup()相同
```