

向量

有序向量：唯一化

$\theta_2 - D_1$

“面壁计划已经恢复，您被指定为唯一的面壁者。”

贾政道：“我要你另换个主意,不许雷同了前人，只做个破题也使得。” 宝玉只得答应着，低头搜索枯肠。

邓俊辉

deng@tsinghua.edu.cn

有序性及其甄别

❖ 还记得**起泡排序**的原理？有序/无序序列中，任何/总有一对**相邻元素**顺序/逆序

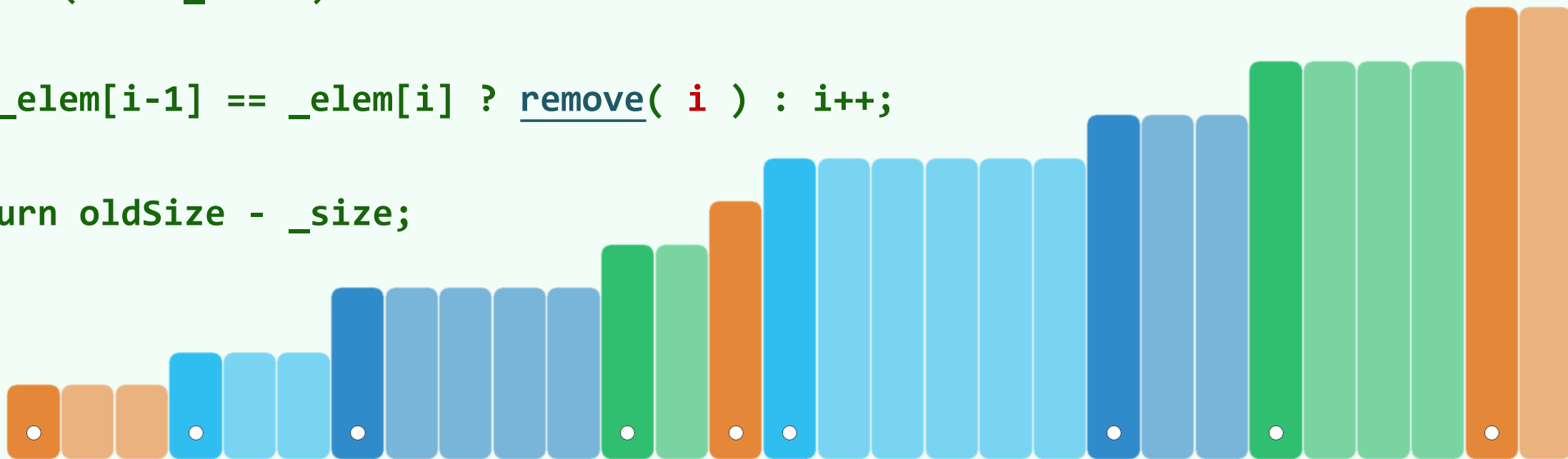
❖ 因此，相邻逆序对的数目，可在一定程度上**度量**向量的**紊乱**程度

```
❖ template <typename T> void checkOrder ( Vector<T> & V ) { //通过遍历
    int unsorted = 0; V.traverse( CheckOrder<T>(unsorted, V[0]) ); //统计紧邻逆序对
    if ( 0 < unsorted )
        printf ( "Unsorted with %d adjacent inversion(s)\n", unsorted );
    else
        printf ( "Sorted\n" );
}
```

❖ 无序向量经**预处理**转换为有序向量之后，相关算法多可优化

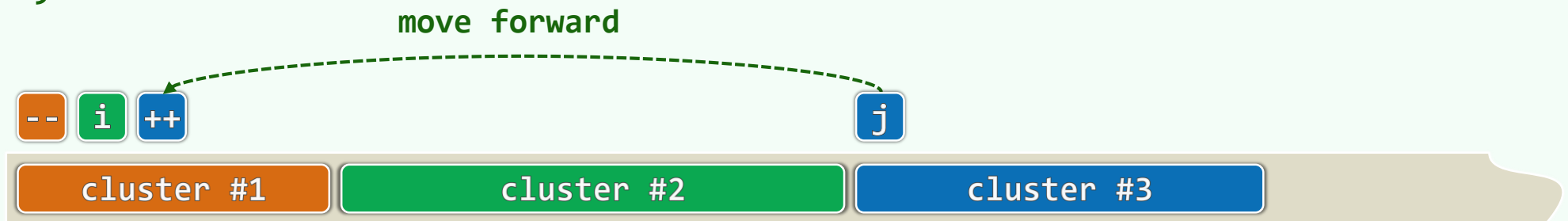
勤奋的低效算法

```
template <typename T> int Vector<T>::uniquify() {  
  
    int oldSize = _size; int i = 1;  
  
    while ( i < _size )  
  
        _elem[i-1] == _elem[i] ? remove( i ) : i++;  
  
    return oldSize - _size;  
  
}
```



懒惰的高效算法

```
template <typename T> int Vector<T>::uniquify() {  
    Rank i = 0, j = 0;  
    while ( ++j < _size )  
        if ( _elem[ i ] != _elem[ j ] )  
            _elem[ ++i ] = _elem[ j ];  
    _size = ++i; shrink();  
    return j - i;  
}
```



实例

