

向量

有序向量：唯一化

“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> int Vector<T>::disordered() const {  
    int n = 0; //相邻逆序对的计数器  
  
    for ( int i = 1; i < _size; i++ ) //逐一检查各对相邻元素  
        n += ( _elem[i-1] > _elem[i] ); //逆序则计数  
  
    return n; //向量有序，当且仅当n = 0  
}
```

❖ 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

低效算法

❖ `template <typename T> int Vector<T>::uniquify() {`

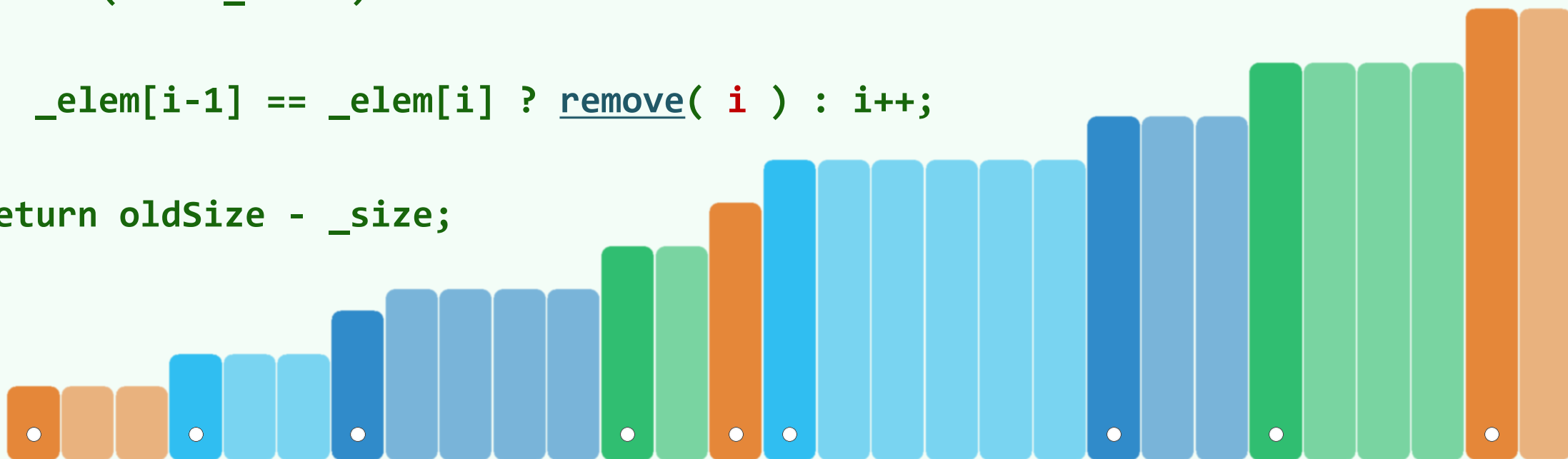
`int oldSize = _size; int i = 1;`

`while (i < _size)`

`_elem[i-1] == _elem[i] ? remove(i) : i++;`

`return oldSize - _size;`

`}`



高效算法

```
❖ template <typename T> int Vector<T>::uniquify() {
```

```
    Rank i = 0, j = 0;
```

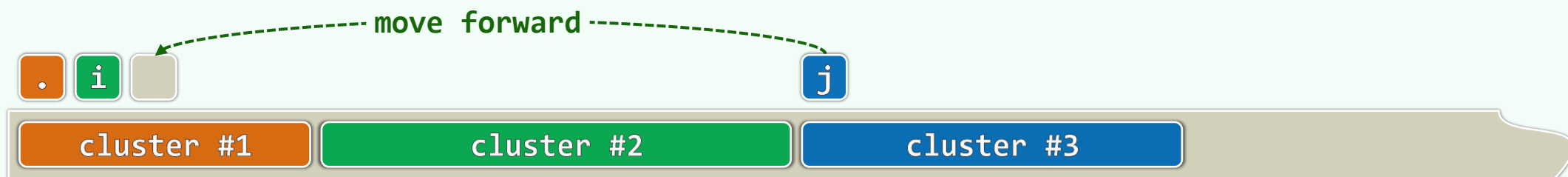
```
    while ( ++j < _size )
```

```
        if ( _elem[ i ] != _elem[ j ] ) _elem[ ++i ] = _elem[ j ];
```

```
    _size = ++i; shrink();
```

```
    return j - i;
```

```
}
```



实例

