

向量

无序向量：查找

02-C2

他便站将起来，背着手踱来踱去，侧眼把那些人逐个个觑将去，
内中一个果然衣领上挂着一寸来长短彩线头

邓俊辉

deng@tsinghua.edu.cn

判等器 ~ 比较器 | 无序 ~ 有序

```
template <typename K, typename V> struct Entry { //词条模板类
    K key; V value; //关键码、数值
    Entry ( K k = K(), V v = V() ) : key ( k ), value ( v ) {}; //默认构造函数
    Entry ( Entry<K, V> const& e ) : key ( e.key ), value ( e.value ) {}; //克隆

    // 被判为相等的词条，未必相同
    bool operator== ( Entry<K, V> const& e ) { return key == e.key; } //等于
    bool operator!= ( Entry<K, V> const& e ) { return key != e.key; } //不等于

    // 得益于比较器和判等器，从此往后，不必严格区分词条及其对应的关键码
    bool operator< ( Entry<K, V> const& e ) { return key < e.key; } //小于
    bool operator> ( Entry<K, V> const& e ) { return key > e.key; } //大于
};
```

无序向量：顺序查找

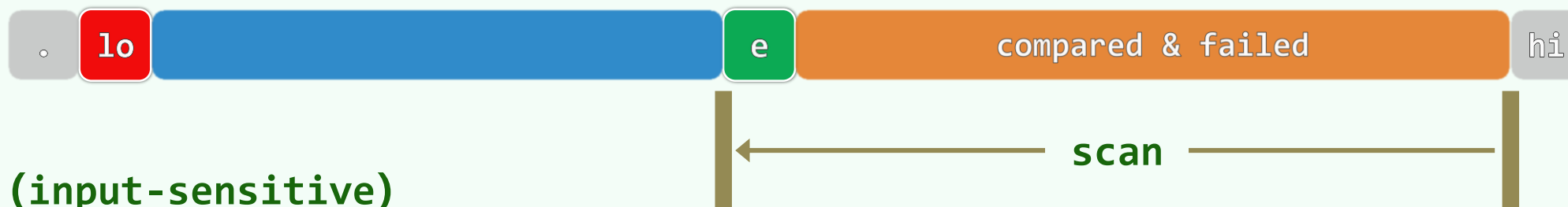
```
template <typename T> //0 <= lo < hi <= _size
```

```
Rank Vector<T>::find( T const & e, Rank lo, Rank hi ) const { //O(hi-lo)
```

```
    while ( (lo < hi--) && (e != _elem[hi]) ); //从后向前，顺序查找
```

```
    return hi; //最靠后的命中者，或lo-1示意失败 (lo == 0时呢? )
```

```
}
```



❖ 输入敏感 (input-sensitive)

最好 $O(1)$ ，最差 $O(n)$ ，有实质区别