

向量

## 有序向量：二分查找（版本A）

e<sub>2</sub> - D<sub>2</sub>

自从爷爷去后，这山被二郎菩萨点上火，烧杀了大半。我们蹲在井里，躲在涧内，藏于铁板桥下，得了性命。及至火灭烟消，出来时，又没花果养赡，难以存活，别处又去了一半。我们这一半，捱苦的住在山中，这两年，又被些打猎的抢了一半去也。

邓俊辉

deng@tsinghua.edu.cn

# 统一接口

❖ template <typename T> //查找算法统一接口， $0 \leq lo < hi \leq _size$

```
Rank Vector<T>::search( T const & e, Rank lo, Rank hi ) const {
```

```
    return ( rand() % 2 ) ? //按各50%的概率随机选用
```

```
        binSearch( _elem, e, lo, hi ) //二分查找算法，或者
```

```
        : fibSearch( _elem, e, lo, hi ); //Fibonacci查找算法
```

```
}
```

*lo*

*hi*



*0*

*\_size*

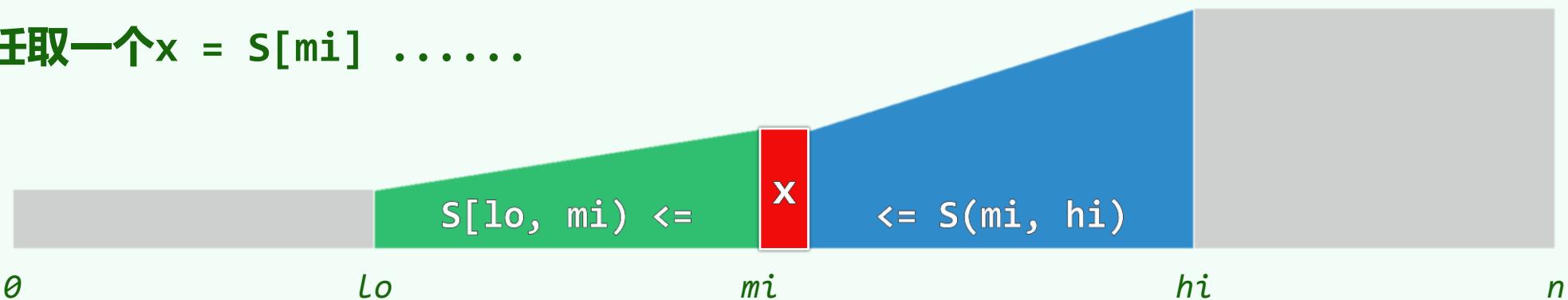
# 轴点

❖ 有序向量中，处处皆为轴点 //说人话！

❖ 以任一元素 $s[mi]$ 为界

- 都可将待查找区间 $[lo, hi)$ 分为三部分，且
- $s[lo, mi] \leq s[mi] \leq s(mi, hi)$

❖ 因此，只需任取一个 $x = s[mi]$  .....



# 减而治之

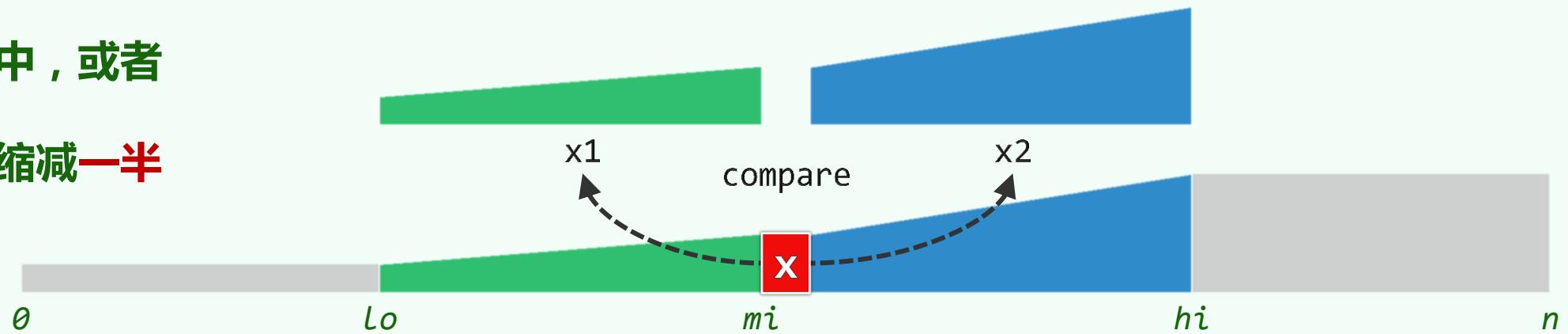
❖ 只需将目标元素 $e$ 与 $x$ 做一比较，即可分三种情况进一步处理：

- $e < x$ ：则 $e$ 若存在必属于**左侧子区间**，故可（减除 $s[mi, hi]$ 并）递归深入 $s[lo, mi]$
- $x < e$ ：则 $e$ 若存在必属于**右侧子区间**，亦可（减除 $s[lo, mi]$ 并）递归深入 $s(mi, hi)$
- $e = x$ ：已在此处**命中**，可随即返回 //若有多个，返回何者？

❖ 若轴点 $mi$ 取作中点，则每经过**至多两次比较**

❖ 或者能够命中，或者

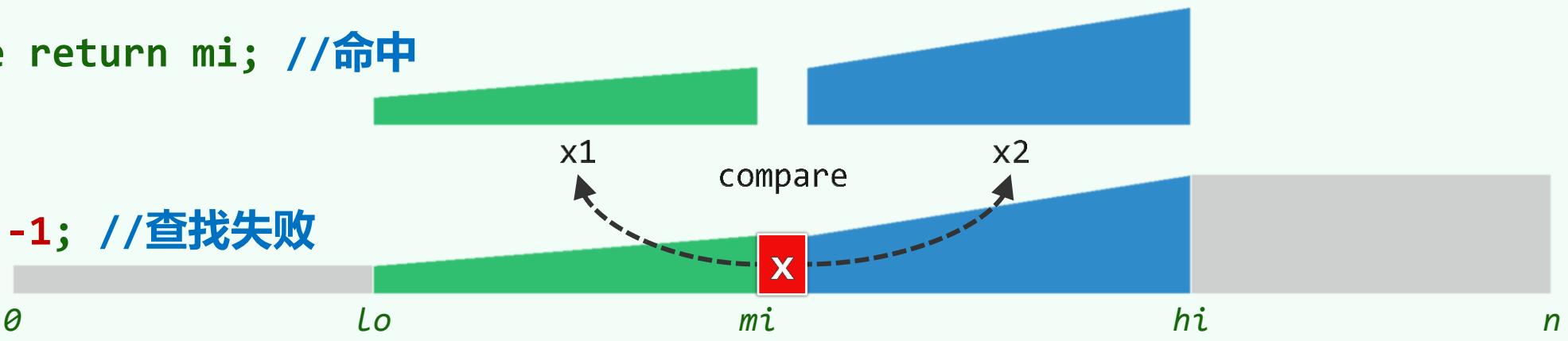
将问题规模缩减一半



# 实现

❖ template <typename T> //在有序向量区间[lo, hi)内查找元素e

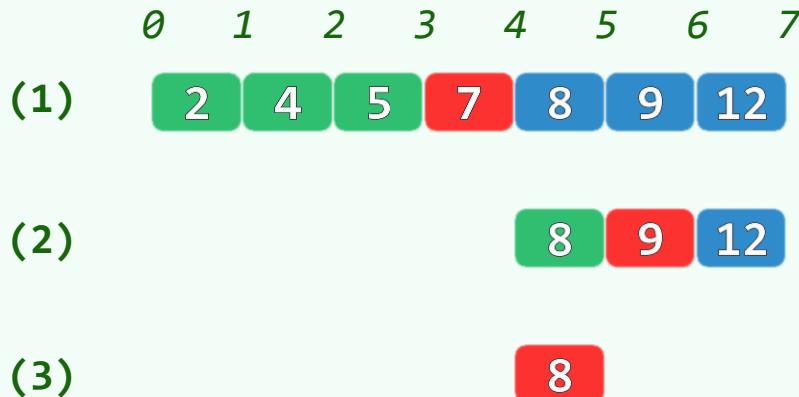
```
static Rank binSearch( T * s, T const & e, Rank lo, Rank hi ) {  
    while ( lo < hi ) { //每步迭代可能要做两次比较判断，有三个分支  
        Rank mi = ( lo + hi ) >> 1; //轴点居中（区间宽度折半，等效于其数值右移一位）  
        if ( e < s[mi] ) hi = mi; //深入前半段[lo, mi)  
        else if ( s[mi] < e ) lo = mi + 1; //深入后半段(mi, hi)  
        else return mi; //命中  
    }  
    return -1; //查找失败  
}
```



## 实例 + 复杂度

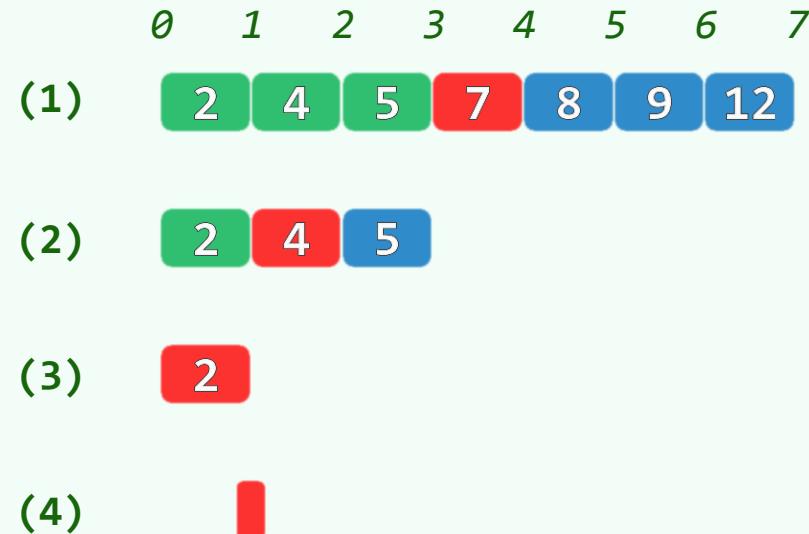
❖ `S.search(8, 0, 7)` :

经  $2 + 1 + 2 = 5$  次比较，在`s[4]`命中



`S.search(3, 0, 7)` :

经  $1 + 1 + 2 = 4$  次比较，在`s[1]`失败

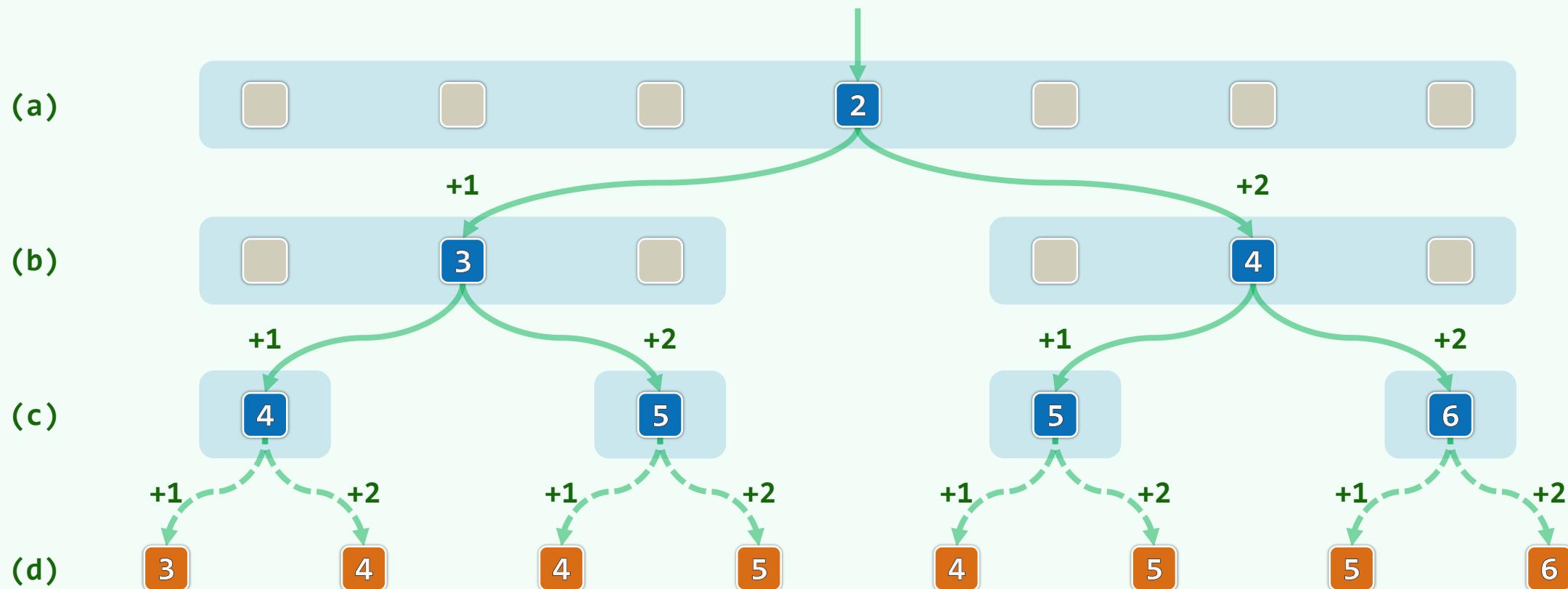


❖ 线性递归： $T(n) = T(n/2) + \mathcal{O}(1) = \mathcal{O}(\log n)$ ，大大优于顺序查找

递归跟踪：轴点总能取到中点，递归深度  $\mathcal{O}(\log n)$ ；各递归实例仅耗时  $\mathcal{O}(1)$

# 查找长度

- 如何更精细地评估查找算法的性能？考查关键码的比较次数 ( search length )
- 通常，需分别针对成功与失败查找，从最好、最坏、平均等角度评估
- 比如，成功、失败时的平均查找长度均大致为  $\mathcal{O}(1.50 \cdot \log n)$  //详见教材、习题解析



# 查找长度

❖  $n = 7$  时，各元素对应的成功查找长度为

{ 4, 3, 5, 2, 5, 4, 6 }

在等概率情况下，平均成功查找长度

$$= 29 / 7 = 4.14$$

❖ 共8种失败情况，查找长度分别为

{ 3, 4, 4, 5, 4, 5, 5, 6 }

在等概率情况下，平均失败查找长度

$$= 36 / 8 = 4.50$$

(a)



(b)



(c)



(d)

