

Teach me half the gladness
That thy brain must know,
Such harmonious madness
From my lips would flow
The world should listen then,
as I am listening now.

2. 向量

有序向量

二分查找 (版本C)

邓俊辉

deng@tsinghua.edu.cn

实现

```
❖ template <typename T> static Rank binSearch( T * A, T const & e, Rank lo, Rank hi ) {  
    while ( lo < hi ) { //不变性:  $A[0, lo) \leq e < A[hi, n)$   
        Rank mi = (lo + hi) >> 1; //以中点为轴点, 经比较后确定深入  
        e < A[mi] ? hi = mi : lo = mi + 1; //[lo, mi)或(mi, hi)  
    } //出口时,  $A[lo = hi]$ 为大于e的最小元素  
    return --lo; //故 $lo - 1$ 即不大于e的元素的最大秩  
}
```

❖ 与版本B的差异

- 待查找区间宽度缩短至 0 而非 1 时, 算法才结束 //lo == hi
- 转入右侧子向量时, 左边界取作 $mi + 1$ 而非 mi //A[mi]会被遗漏?
- 无论成功与否, 返回的秩严格符合接口的语义约定... //如何证明其正确性?

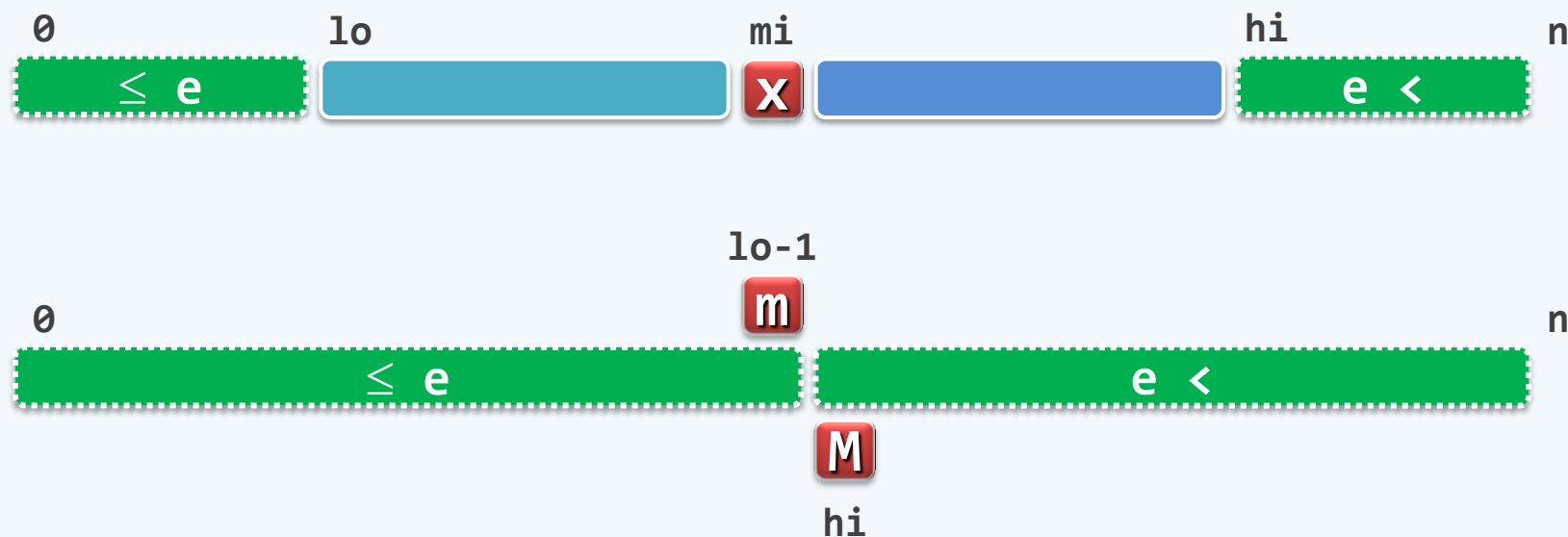
不变性 : $A[lo, lo) \leq e < A[hi, n)$

❖ 在算法执行过程中的任意时刻

$A[lo - 1] / A[hi]$ 总是 (截至当时已确认的) 不大于e的最大者 / 大于e的最小者

❖ 当算法终止时 ($lo == hi$)

$A[lo - 1] / A[hi]$ 即是 (全局) 不大于e的最大者 / 大于e的最小者

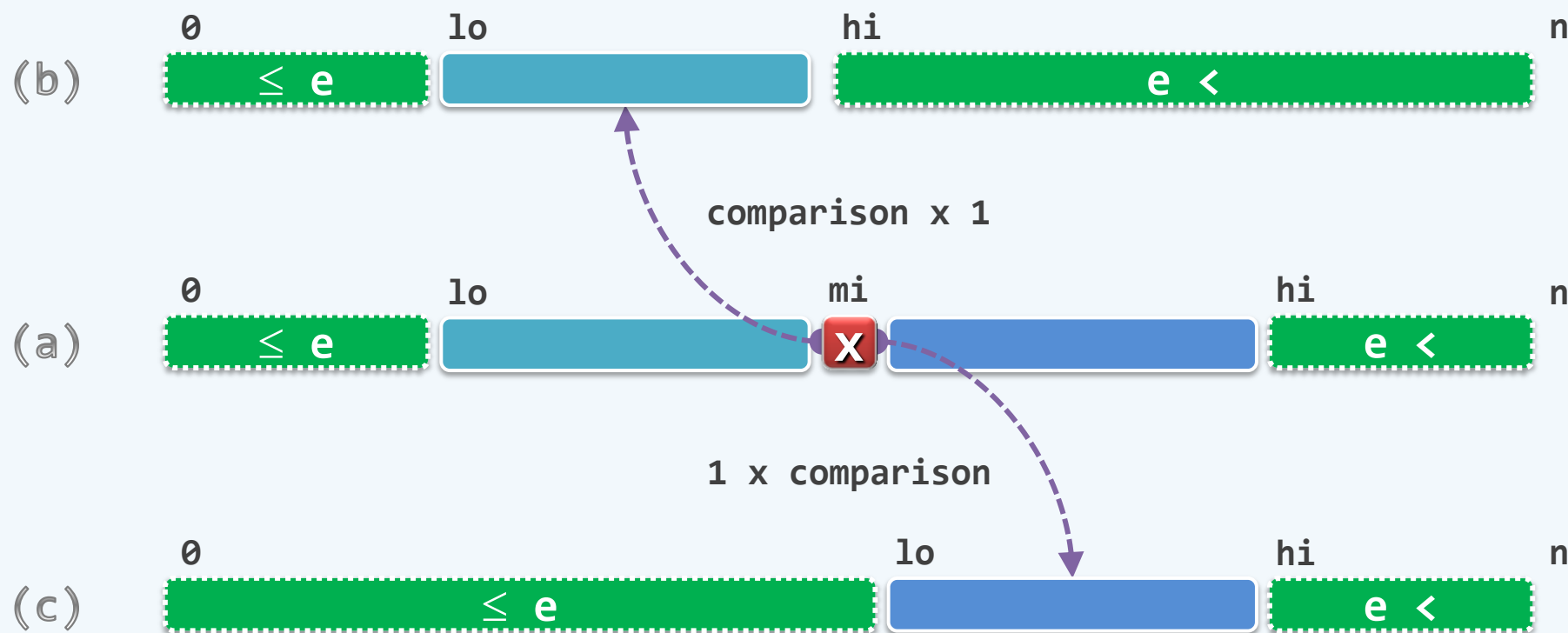


不变性 : $A[0, lo) \leq e < A[hi, n)$

❖ 初始时, $lo = 0$ 且 $hi = n$, $A[0, lo) = A[hi, n) = \emptyset$, 自然成立

❖ 数学归纳 : 假设不变性一直保持至 (a)

//以下无非两种情况...



❖ 针对二分查找，Knuth (ACP-v3-s6.2.1-ex_23) 曾指出：

将三分支变为两分支后的改进效果

需要到n非常大 ($2^{2*(17-(-16))} = 2^{66}$) 后方能体现

试阅读相关段落；这一结论，对当下的实际应用有何意义？