

## 02-D4-1 构思 & 02-D4-2 版本B

### #数据结构邓神

二分查找中的左右分支转向代价不平衡的也可以直接解决

为什么我们不能讲左右转向的代价平衡呢？

每次迭代（或者每个递归实例），只做一次关键码比较

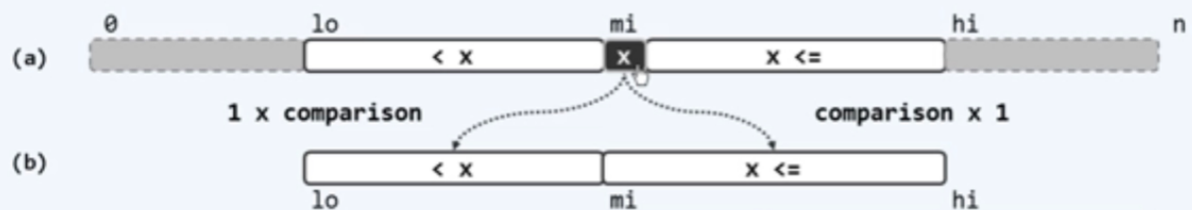
如此所有分支只有两个方向，而不是三个分支

❖ 同样地，轴点 $mi$ 取作中点，则查找每深入一层，问题规模也缩减一半

1)  $e < x$  : 则 $e$ 若存在必属于左侧子区间 $s[lo, mi)$ ，故可递归深入

2)  $x \leq e$  : 则 $e$ 若存在必属于右侧子区间 $s[mi, hi)$ ，亦可递归深入

只有当元素数目 $hi - lo = 1$ 时，才判断该元素是否命中



我们判断目标关键码是否严格小于节点

如果是  $\geq$  就直接向右侧深入（我们将取等号的算法归入到右侧，这会导致我们不能及时判定是否命中，只有当区间长度缩减为1到时候才能判断是否命中）

```
template <typename T> static Rank binSearch(T* A, T const& e, Rank lo, Rank hi){
    while (1 < hi - lo){
        Rank mi = (lo + hi) >> 1;
        (e < A[mi])? hi = mi: lo = mi;
    }
    return e==A[lo]?lo:-1;
}
```

相对于之前的版本，最好的情况性能会下降，最坏的情况会更好，各种情况下的SL更加接近，整体性能会更

加趋于稳定。