

二分查找问题

1. 问题描述

- 描述：在子表 $L[start:end]$ 中查找 E ，并返回 E 在子表中的位置
- 输入： $L, start=1, end=n, E$
- 输出：若 E 在 L 中存在，返回位置，否则返回-1。

2. 算法思想

递归，分治

3. 步骤

3.1 自然语言描述

1. 若 $L[start:end]$ 为空，则返回 -1.
2. 比较 E 与子表的中间位置 $(start + end) / 2$ 的元素 m 进行比较
3. 若相等则返回位置 $(start + end) / 2$
4. 若 E 小于 m 则用相同的方法查找 $L[start: (start + end) / 2 - 1]$
5. 若 E 大于 m 则用相同的方法查找 $L[(start + end) / 2 + 1: end]$

3.2 伪代码描述

```
//在L的子表L[start:end]中查找E
binarySearch(L, E, start, end)
    if start > end
        return -1;
    middle = (start + end)/2;
    if E == L[middle]
        return middle;
    if E < L[middle]
        return binarySearch(L, E, start, middle - 1);
    if E > L[middle]
        return binarySearch(L, E, middle + 1, end);
```