

# 二分搜索代码要点

- 对输入做异常处理：数组为空或者数组长度为0。
- $\text{int mid} = \text{start} + (\text{end} - \text{start}) / 2$  这种表示方法可以防止两个整型值相加时溢出。
- Recursion or While-Loop：使用迭代而不是递归进行二分查找，因为工程中递归写法存在潜在溢出的可能
- while循环终止条件：while终止条件应为 $\text{start} + 1 < \text{end}$ 而不是 $\text{start} \leq \text{end}$ ， $\text{start} == \text{end}$ 时可能出现死循环，即循环终止条件是相邻或相交元素时退出。配合while终止条件 $\text{start} + 1 < \text{end}$ （相邻即退出）的赋值语句mid永远没有+1或者-1，这样不会死循环。
- 迭代终止时target应为start或者end中的一个。循环终止条件有两个，具体应看是找第一个还是最后一个而定。

# 为什么不写成 $\text{start} \leq \text{end}$

- input: [3,4,5,8,8,8,8,10,13,14], 找到第一个“8”出现的位置
- output: 4
- expect: 3