

HW2

范潇 2254298

2024 年 4 月 17 日

题目 1. (主元素问题) 设 $T[0..n-1]$ 是 n 个元素的数组。对任一元素 x ，设 $S(x) = \{i | T[i] = x\}$ 。当 $|S(x)| > n/2$ 时，称 x 为 T 的主元素。

1. 如果 T 中元素是有序的，按分治策略设计并实现一个线性时间算法，确定 $T[0..n-1]$ 中是否有主元素；
2. 若 T 中元素不是有序的，试设计并实现一个 $O(n \log n)$ 有效算法，确定 T 是否有主元素。进一步，能找到一个线性时间算法吗？

解答.

1. 因为 $n - \lceil n/2 \rceil \leq \lceil (n+1)/2 \rceil \leq \lceil (n+1)/2 \rceil$ ，所以若 T 存在主元素，则 $T[\lceil (n+1)/2 \rceil]$ 一定等于该主元素。因此只需要根据定义判断 $T[\lceil (n+1)/2 \rceil]$ 是否为主元素即可。由于 T 保证是有序的，从而相同的元素连在一起，可以利用二分查找来获取值为 $T[\lceil (n+1)/2 \rceil]$ 的元素下标的最值，从而可以直接得到对应的元素数量，然后再根据定义进行判断即可。伪代码在第 2 页中给出。时间复杂度为 $\Theta(\log n)$ 。
2. 若 T 不保证有序，只需要先对其排序后再输入至 `checkPivotElementRecursively` 即可得到 $O(n \log n)$ 的有效算法，伪代码在第 2 页中给出。如果想要进一步得到线性时间算法，可以使用线性选择函数来直接从 T 中选取第 $\lceil (n+1)/2 \rceil + 1$ 大的元素，从而避免了排序，使得时间复杂度降为 $\Theta(n)$ ，伪代码在第 3 页中给出。

Procedure checkPivotElementRecursively(T)

Input: 升序数组 T
// 下标从 0 开始

```
1 pivot =  $T[(n+1)/2]$ 
2  $l = 0$ 
3  $r = n-1$ 
4 while  $l < r$  do
5   |    $mid = \lfloor (l+r)/2 \rfloor$ 
6   |   if  $T[mid] \geq pivot$  then  $r = mid$  else  $l = mid+1$ 
7 end while
8  $left = l$ 
9  $l = 0$ 
10  $r = n-1$ 
11 while  $l < r$  do
12   |    $mid = \lfloor (l+r+1)/2 \rfloor$ 
13   |   if  $T[mid] \leq pivot$  then  $l = mid$  else  $r = mid-1$ 
14 end while
15  $right = r$ 
16 if  $right-left+1 > n/2$  then
17   |   return True
18 else
19   |   return False
20 end if
```

Algorithm 1: checkPivotElement(T)

Input: 任意数组 T

```
1 Sort ( $T$ )// 任意高效排序算法, 如堆排序
2 return checkPivotElementRecursively ( $T$ )
```

Algorithm 2: linearCheckPivotElement(T)

Input: 任意数组 T

// 下标从 0 开始

```
1 pivot = linearSelect ( $T, \lceil (n+1)/2 \rceil + 1$ ) // 选取第  $\lceil (n+1)/2 \rceil + 1$  大的元素
2 cnt = 0
3 foreach  $x$  in  $T$  do
4   | if  $x == pivot$  then cnt += 1
5 end foreach
6 if  $cnt > n/2$  then
7   | return True
8 else
9   | return False
10 end if
```
