

2-9:

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 $|S(x)| > \frac{n}{2}$ 时，称 x 为 T 的主元素。设计一个线性时间算法，确定 $T[0:n-1]$ 是否有一个主元素。

题目分析：

$|S(x)|$ 表示数组 T 中等于 x 的元素个数，若 $|S(x)| > \frac{n}{2}$ ，则必有一结论： x 为数组 T 的中位数（此处假设数组为偶数时，中位数取为第 $\frac{n}{2}$ 个数）。

证明：若 n 为偶数，则 $|S(x)| > \frac{n}{2} \geq \frac{n}{2} + 1$ ，考虑极端情况： x 为 T 中最大的元素，则 x 至少会排到第 $\frac{n}{2}$ 个，即 x 为中位数；若 n 为奇数，第 $\frac{n}{2}$ 个元素必然为 x ，即 x 为中位数。

所以，此题可先用线性时间选择选出原数组中的中位数，再遍历数组，确认其是否为主元素即可，复杂度应为 $O(n)$ 。

代码实现：

详见2-9 Code.cpp

输出示例：

```
1  输入样例1:
2  16
3  3 9 8 6 11 21 49 38 5 4 7 1 2 15 23 8
4  输出1: 不存在
5  输入样例2:
6  16
7  2 3 5 2 1 8 2 3 2 2 2 2 5 2 3 2
8  输出2: 存在（9个2）
9  输入样例3:
10 7
11 3 1 2 3 3 7 3
12 输出3: 存在（4个3）
13 输入样例4:
14 7
15 1 7 3 2 3 3 1
16 输出4: 不存在（3个3）
```

算法分析：

线性时间选择中位数复杂度为 $O(n)$ （证明略）

遍历数组记数复杂度显然为 $O(n)$

因此，总算法复杂度为 $T(N) = O(n) + O(n) = O(n)$