

# 算法设计第一次作业

---

2\_7

**Answer:**

由已知条件得，算法可在  $O(i \log i)$  时间内计算两个  $i$  次多项式的乘积。

且原式可表示为：

$$P(x) = (x - n_1)(x - n_2) \dots (x - n_d)$$

则由分治思想，可将  $d$  次多项式转换为2个  $d/2$  次多项式的乘积，则可设计算  $d$  次多项式的时间为  $T(d)$ ，则可表示为：

$$T(d) = \begin{cases} O(1), & d = 1 \\ 2T(d/2) + O(d \log d), & d > 1 \end{cases}$$

则对此递归求解即可得：

$$T(d) = O(d \log^2 d).$$

2\_9

**Answer:**

将此题解释即可分析出，此题希望求出函数一个解所占的整个解空间的规模大于一半，即可称为主元素。

则由此可得出结论， $x$  必为排好序的数组  $T$  的中位数。

易证明：

$S(x) > n/2$  时，则按照序关系排好后，必有中位数被主元素占据。

则只需要在线性时间中找出数组的中位数，再遍历查看中位数是否与主元素相同即可。则复杂度应为  $O(n)$

详情代码可见github网址：[https://github.com/vegetablechickenluo/design\\_algorithm/blob/master/work2\\_9/main.cpp](https://github.com/vegetablechickenluo/design_algorithm/blob/master/work2_9/main.cpp)

输出样例：

请输入数组长度n: 5

请输入依次输入n个数: 2 3 4 4 3

不存在

请输入数组长度n: 3

请输入依次输入n个数: 1 1 2

存在

请输入数组长度n: 6

请输入依次输入n个数: 4 3 3 4 4 4

存在

此题易得出: 时间复杂度为  $O(n)$