算法设计第一次作业

2_7

Answer:

由已知条件得，算法可在 *O(i logi)* 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：

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

则对此递归求解即可得：

$$T(d) = O(dlog^2 d).$$

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)*

详情代码可见本人的github网址：(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)*

2_10

Answer:

由于此题目中数组 **T** 的元素不再具有序关系，因此不可以用 *2_9* 中线性时间寻找中位数的方法去寻找。

换用二分的思想来尝试解决该问题，假设[0 : (n - 1) /2] 部分与[(n - 1) / 2 : (n - 1)] 部分的主元素不为x，则合并后的整个数组的主元素不为x。因此，可以采用二分的思想来解决该问题。

则算法的递归式表示为：

$$T(n) = 2T(n/2) + O(n)$$

由《算法导论》中的主定理，易求出算法复杂度下界为：*O(n logn)*

详情代码可见本人的github网址——*由于此题数组设定为无序，则不会采用比较的方式*：

（https://github.com/vegetablechickenluo/design_algorithm/blob/master/work2_10/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,主元素是1

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，主元素是4
```

算法优化：

不妨采用遍历的方式来进行比较，当x占据整个数组大于 *n / 2* 范围时，则可认为x为主元素，因此可以开n * n的二维数组作为储存每一个数字的空间。*a [n] [n]* 中每一行代表每一个不同的数，每一列用0或1表示占据的次数，最终通过遍历求和的方式来求的每一个数出现的次数，最终与 *n / 2*比较，则 *O(n)* 为优化后的时间复杂度下界。

2_28

Answer：

注：已排好序默认为升序排列；

题目分析：

题目要求设计出 *O (log n)* 时间复杂度的算法，则首先则需要对题目中的两个数组利用分治思想进行处理。

算法表示：

I. 分别寻找长度为 *n* 的 X、Y两个数组的中位数，由于两数组已排好序，则查找的时间复杂度为 *O(1)*；

II. 然后将 *X*、*Y* 两数组的中位数设为 x1,y1, 比较 x1 与 y1 的大小关系，时间复杂度为 *O(1)*；

III. **important**：依据中位数，将 *X*、*Y* 两个数组依次分为 *X1, X2* 与 *Y1, Y2* 四个数组，**则定有中位数小的数组的左半部分，与中位数大的数组的右半部分中不包含中位数**。因此，则可以将其砍去，以实现简单的问题总规模减半的目的。

复杂度分析：

$$T(n) = T(n/2) + O(1)$$

则易得出，算法时间复杂度为 *O(log n)*；

详情代码可见本人的github网址：

（https://github.com/vegetablechickenluo/design_algorithm/blob/master/work2_28/main.cpp）

输出样例：

```
请输入数组规模： 10
请依次输入a数组的数字： 1 2 3 4 5 6 7 8 9 10
请依次输入b数组的数字： 2 3 4 5 6 7 8 9 10 11
中位数为： 6

请输入数组规模： 10
请依次输入a数组的数字： 1 2 3 4 6 6 7 8 9 10
请依次输入b数组的数字： 2 3 4 5 7 7 8 9 10 11
中位数为： 6.5

请输入数组规模： 3
请依次输入a数组的数字： 1 2 3
请依次输入b数组的数字： 2 3 4
中位数为： 2.5
```