2-1

n=1或n=2时递归算法和非递归算法产生相同移动序列

设 k≤n-1时，2种算法产生完全相同的移动序列

将移动分为顺时针 C、逆时针 CC、非最小圆盘塔座间

移动 O

n为奇数时，顺时针非递归算法移动序列为 C，O，C，O…，C；

逆时针非递归算法产生移动序列为 CC，O，CC，O… CC

n为偶数时，顺时针非递归算法序列为 CC，O，CC，O，…CC，

逆时针非递归算法产生移动序列为 C，O，C，O… C

(1) n为奇数时，顺时针递归算法移动序列：

hanoi(n-1，A，C，B)的移动序列，O，hanoi(n-1，C，B，A)序列

可以知道二者为偶数圆盘逆时针移动问题。

由数学归纳法知，移动序列为 C，O，C，O… C。因此，

hanoi(n，A，B，C)产生的移动序列为 C，O，C，O… C

(2) n为偶数时，顺时针递归算法移动序列：

hanoi(n-1，A，C，B)序列，O，hanoi(n-1，C，B，A)序列

且为奇数圆盘逆时针移动问题。

由数学归纳法，序列为 CC，O，CC，O… CC

因此 hanoi(n，A，B，C)产生序列为 CC，O，CC，O… CC

n为奇数和偶数的逆时针递归算法也类似。

故递归算法和非递归算法产生相同序列

2-7

由 $p(n_1)=p(n_2)=\cdots p(n_d)=0$

最 $p(x)$ 为最高次项系数为1的 d 次多项式

有 $(x-n_1)(x-n_2)\cdots(x-n_d)=p(x)=\prod_{i=1}^{d}(x-n_i)$

从而 $p(x)=\prod_{i=1}^{\lfloor \frac{d}{2} \rfloor}(x-n_i)\cdot\prod_{i=\lfloor \frac{d}{2} \rfloor+1}^{d}(x-n_i)=p_1(x)\cdot p_2(x)$

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

解之得：$T(d)=O(d\log^2 d)$

2-8

使用二分查找方法

由于n个整数各不相同，故 $\forall i$ 有 $T[i]\le T[i+1]-1$

对 $1<i\le n-1$，当 $T[i]<i$ 时，对 $1\le j<n$ 有

$T[j]\ge T[i]+j-i>i+j-i=j$

对 $1<i\le n$，当 $T[i]<i$ 时，对 $1\le j<i$ 有

$T[j]\le T[i]-i+j=j$

故可用二分搜索算法在 $O(\log n)$ 时间内找到下标