



**算法设计与分析实验报告**

学院：计算机科学学院

班级：计算机科学与技术1701班

学号：41709010126

姓名：寇明珠

汉诺塔(Hanoi)问题：古代有一个梵塔，塔内有3个基座，A基座上有64个盘子，盘子大小不等，大的在下，小的在上。有一个老和尚想把盘子由A座移到B座，但每次只能移动一个盘子，3个基座上的盘子都始终保持大的在下，小的在上。移动过程中可以利用C基座做辅助，求解其移动过程。

     汉诺塔问题是递归算法比较经典的例题，几乎是每个老师讲解递归算法时都会讲到的问题，对于任意n阶汉诺塔问题，假设有3个基座A，B， C。在基座A上放置有n个直径大小各不同，从小到大依次编号为：1、2、3…n。移动时遵循以下规则：

       1）每次只能移动一个盘子。

       2）盘子可以放置在A、B、C、中的任何一个基座上。

       3）任何时候都不能将一个较大的盘子放在较小的盘子上。

递归算法：

#include<stdio.h>

void hanoi(int n, char a, char b, char c)

{

if (n == 1)

{

printf("\t%c->%c\n", a, b);

}

else

{

hanoi(n - 1, a, c, b);

printf("\t%c->%c\n", a, b);

hanoi(n - 1, c, b, a);

}

}

int main()

{

int n;

printf("请输入要移动的块数: ");

scanf("%d", &n);

hanoi(n, 'a', 'b', 'c');

}

非递归算法：

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdlib.h>

#include<stdio.h>

int main()

{

long long i, res, t, k; int n, s;

scanf("%d", &n);

res = (1 << n) - 1;

printf("%lld\n", res);

for (i = 1; i <= res; i++){

for (t = 2, s = 1; s <= n; s++, t \*= 2)if (i%t == t / 2) break;//i%t=t/2 找 第i步移动的S号盘

k = i / t;//获得第S盘 第几次移动

if (n % 2 == s % 2){// 逆时针

if ((k + 1) % 3 == 0) printf("%d from B to A\n", s);

if ((k + 1) % 3 == 1) printf("%d from A to C\n", s);

if ((k + 1) % 3 == 2) printf("%d from C to B\n", s);

}

else{// 逆时针

if ((k + 1) % 3 == 0) printf("%d from C to A\n", s);

if ((k + 1) % 3 == 1) printf("%d from A to B\n", s);

if ((k + 1) % 3 == 2) printf("%d from B to C\n", s);

}

}

system("pause");

return 0;

}