

## 1.汉诺塔问题:

- 相传在古印度圣庙中,有一种被称为汉诺塔(Hanoi)的游戏。该游戏是在一块铜板装置上,有三个底座(编号 A、 B、 C),在 A 底座自下而上、由大到小按顺序放置 64 个金盘。
- 游戏的目标:把 A 底座上的金盘全部移到 C 底座上,并仍保持原有顺序叠好。
- 操作规则:每次只能移动一个盘子,并且在移动过程中三个底座上都始终保持大盘在下,小盘在上,操作过程中盘子可以置于 A、 B、 C任一底座上。

编写函数,接受一个表示盘子数量的参数和分别表示源、目标和临时底座编号的参数,输出详细的移动步骤(如: "The k times move: from X to Y",X、 Y 代表底座编号)和移动后底座上盘子的分布情况。

给出初值,调用函数,打印输出移动过程。试试盘子数量 num=64?

提示:

1. 使用递归;
2. 对所有盘子编号,使用列表表示底座上存在的盘子的情况。

```
#Hanoi
count=0
def hanoi(n,a,b,c):
    global count
    if n is 1:
        count=count+1
        print("The",count,"times move: from",a," to ",c)
        change(a,c)
        print("Dishes on the A column:",A)
        print("Dishes on the B column:",B)
        print("Dishes on the C column:",C)
    else:
        hanoi(n-1,a,c,b)
        hanoi(1,a,b,c)
        hanoi(n-1,b,a,c)

def change(a,b):
    global A,B,C
    if a is 'A':
        if b is 'B':
            B.append(A[0])
            A.remove(A[0])
        if b is 'C':
            C.append(A[0])
            A.remove(A[0])
    if a is 'B':
        if b is 'A':
            A.append(B[0])
            B.remove(B[0])
        if b is 'C':
            C.append(B[0])
            B.remove(B[0])
    if a is 'C':
        if b is 'B':
            B.append(C[0])
            C.remove(C[0])
        if b is 'A':
            A.append(C[0])
```

```

        C.remove(C[0])
    A=sorted(A)
    B=sorted(B)
    C=sorted(C)
num=input("Please input the initial number of dishes:")
num=int(num)
A=[i for i in range(1,num+1)]
B=[]
C=[]
hanoi(num, 'A', 'B', 'C')

```

2.编写函数,实现插入排序算法。并构造数据,调用函数进行排序。

```

#Insert Sort
def Insert_Sort(index,x):
    index+=1
    if(index>=len(x)):
        return x
    g=(i for i in range(0,index))
    for i in g:
        if(x[i]>x[index]):
            temp=x[index]
            x.remove(x[index])
            x.insert(i,temp)
    x=Insert_Sort(index,x)
    return x

x=input('Please input a list format for unsorted array:')
x=eval(x)
index=0
Insert_Sort(index,x)
print(x)

```

编写函数,利用例题 5-8 的思想实现一种排序算法。并构造数据,调用函数进行排序。

```

#Quick Sort
def Quick_Sort(x):

    if len(x) is 1:
        return x
    elif len(x) is 0:
        return x
    else:
        select=x[len(x)-1]
        left_arr=[]
        right_arr=[]
        for i in range(0,len(x)-1):
            if x[i]<select:
                left_arr.append(x[i])
            else:
                right_arr.append(x[i])
        ans=[]
        ans.extend(Quick_Sort(left_arr))
        ans.append(select)

```

```
    ans.extend(Quick_Sort(right_arr))  
    return ans  
  
x=input('Please input a list format for unsorted array:')  
x=eval(x)  
index=0  
print(Quick_Sort(x))
```