

遞迴好處：

把大問題拆成小問題解決

河內塔另一種不用遞迴的寫法

```
int main() {
    long long int s,n,a,b;
    typedef pair<int,int> move;
    move temp;
    vector<move> moves;
    cin>>n;
    int start=1,side=2,t=3;
    int starts=1,sides=2,ts=3;
    if(n%2==0)
    {
        swap(side,t);
        swap(t,start);
    }
    for(int i1=0;i1<n;i1++)
    {
        starts=start,sides=side,ts=t;

        s=moves.size();
        if(start==1)
        {
            swap(side,t);
            swap(t,start);
        }
        else
        {
            swap(side,start);
            swap(t,start);
        }
        // printf("%d %d %d\n",start,side,t);
        moves.push_back(make_pair(1,side));
        for(int i=0;i<s;i++)
        {
            temp=moves[i];
```

```

        if(temp.first==starts)temp.first=start;
        else if(temp.first==sides)temp.first=side;
        else if(temp.first==ts)temp.first=t;
        if(temp.second==starts)temp.second=start;
        else if(temp.second==sides)temp.second=side;
        else if(temp.second==ts)temp.second=t;
        moves.push_back(temp);
    }

}
cout<<pow(2,n)-1<<"\n";
for(auto i:moves)cout<<i.first<<" "<<i.second<<"\n";
}

```

資料抽象化目的目的:

更容易讀

更容易寫

更容易修改

const 表示為常數(不可修改)

子類別繼承父類別

多載可以透過傳遞的資料型態來判斷確切為哪個函數

鏈結串列:

插入(刪除)時不需要移動後面的資料