C/C++程序设计案例实战
——汉诺塔游戏

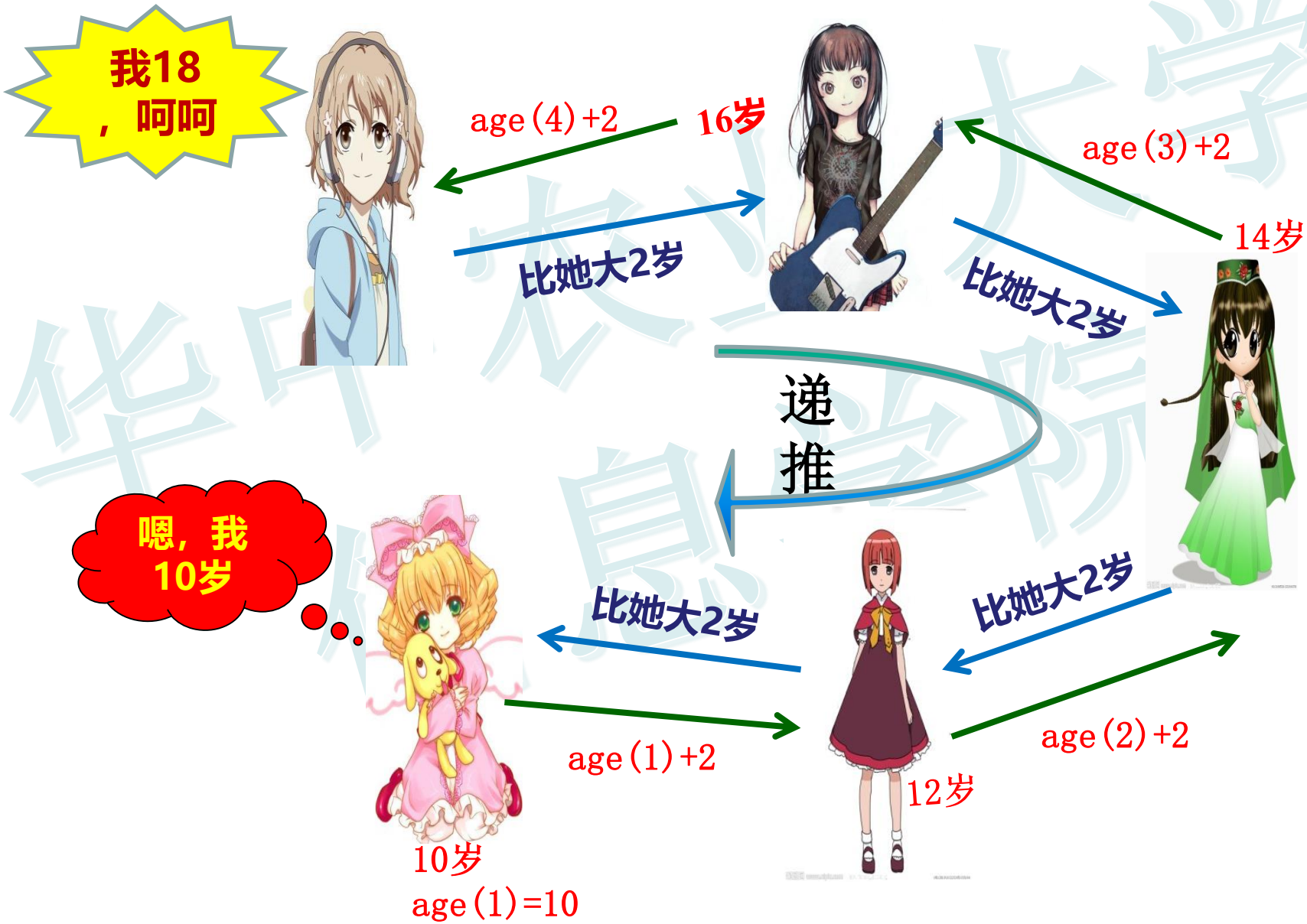
华中农业大学信息学院　李小霞

德罗斯特效应

盗梦空间

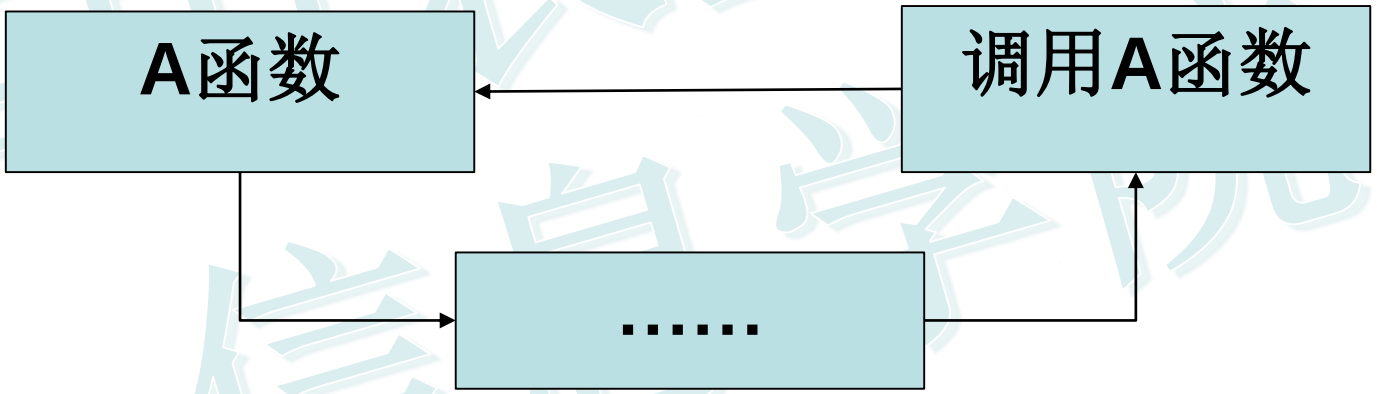
从前有座山，山里有个庙，
庙里有个老和尚
给小和尚讲故事，讲的那是
从前有座山，
山里有个庙，
庙里有个老和尚
给小和尚讲故事
讲的那是
从前有座山...

案例引入——猜年龄


我18，呵呵
age(4)+2
16岁
比她大2岁
age(3)+2
14岁
比她大2岁
递推
嗯，我10岁
比她大2岁
比她大2岁
age(1)+2
age(2)+2
12岁
10岁
age(1)=10

递归调用定义：

一个函数在它的函数体内直接或间接**调用它自身**称为递归调用，这种函数称为递归函数。

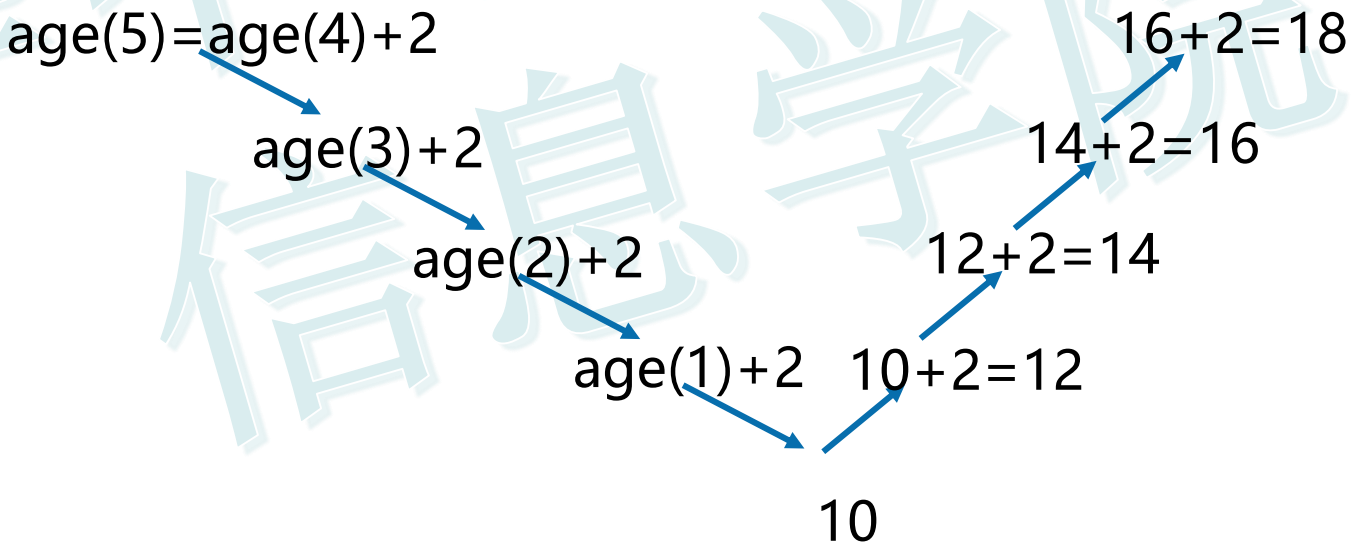


递归函数：

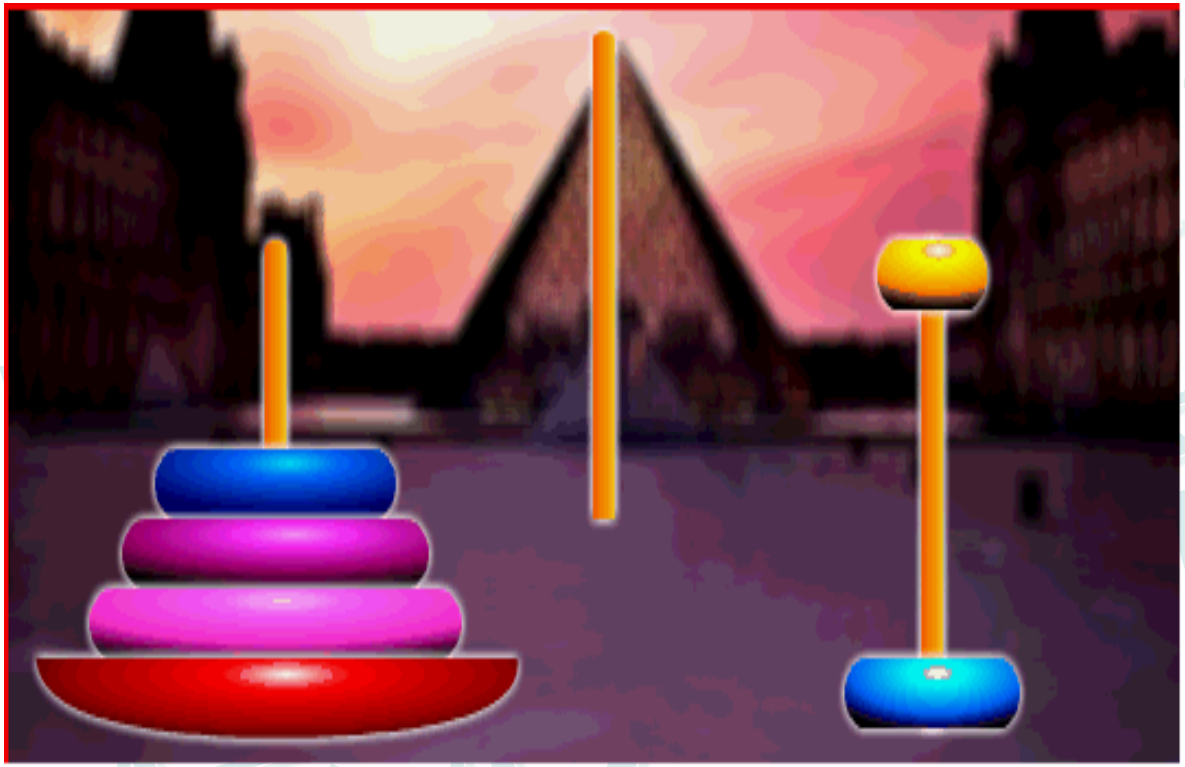
```
返回值类型  函数名（形参列表）
{
        ……
        if(条件)
            递归结束；
        else
            递归调用；
}
```

```
1 #include<iostream>
2 using namespace std;
3 int age(int n){
4  if(n==1)
5    return 10;
6  else
7    return age(n-1);
8  }
```

```
9  int main(){
10   int age5;
11   age5=age(5);
12   cout<<"the 5th girl's age
13          is "<<age5<<endl;
14 }
```



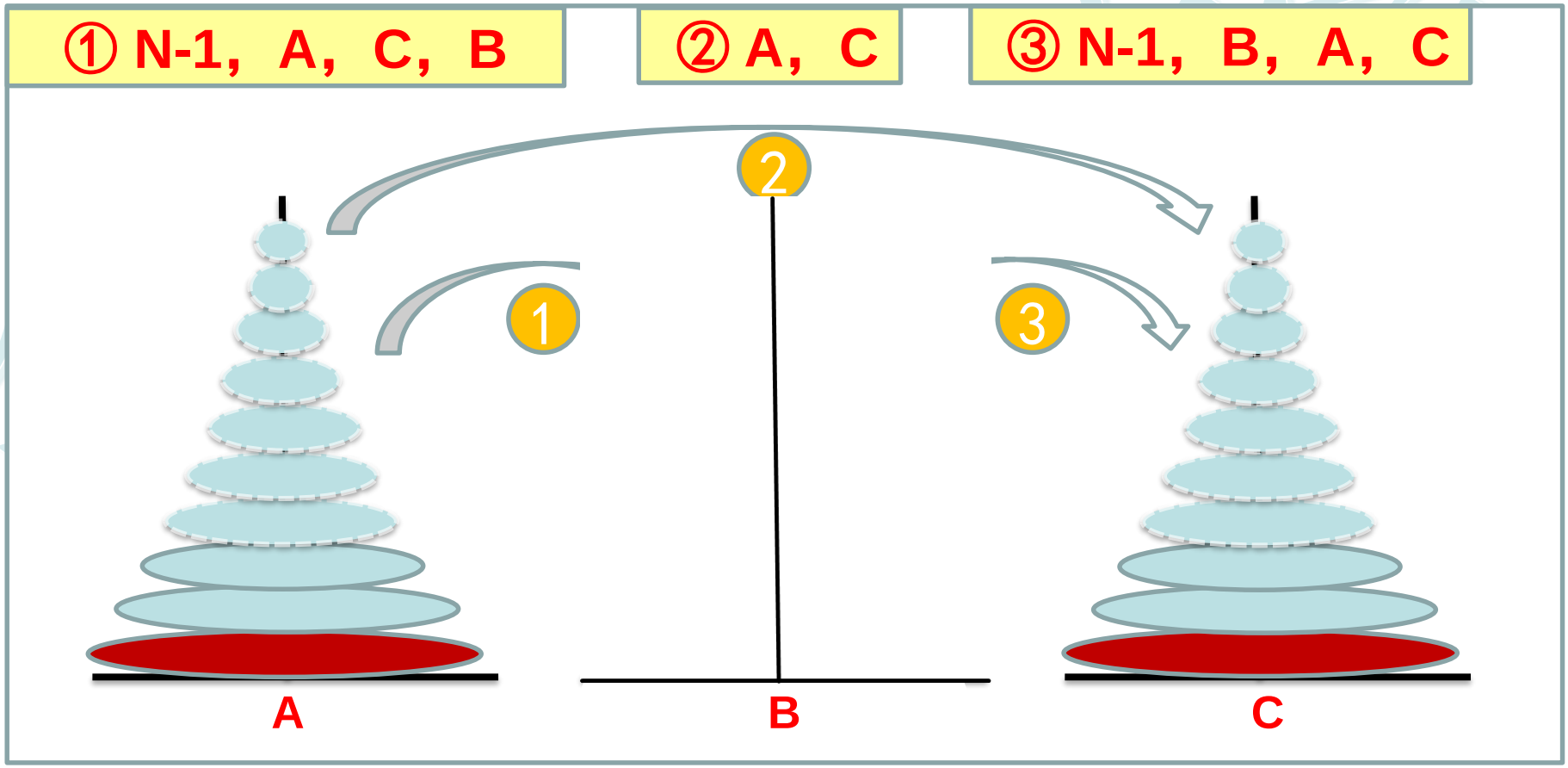
汉诺塔（Hanoi）问题



有人曾计算过，当n=64时，所需移动的次数为18446744073709551615，即**1844亿亿次**

若按每次耗时1微秒计算，则完成64个圆盘的移动将需要60万年

用递归求解汉诺塔



代码实现

移动一个盘子
（递归出口）

```
void hanoi(int n,char a,char b,char c){
    if(n==1)    move(n,a,c);
    else{
            hanoi(n-1,a,c,b);
            move(n,a,c);
            hanoi(n-1,b,a,c);}
}
```

借助b盘将n-1个盘子从a移到c（递归调用）

代码实现

```
void move(char in,char out){
    cout<<n<<" "<<in<<"->"<<out<<endl;
}
```

```
int main()
{
    int n;
    cin>>n;
    cout<<"The steps to move "<<n<<"
          disks"<<endl;
    hanoi(n,'a','b','c');
    return 0;
}
```

小结

递归函数

延申

请应用递归函数解决如下问题：

一对兔子每个月能生出一对小兔子。如果所有兔子都不死，那么一年以后可以繁殖多少对兔子?