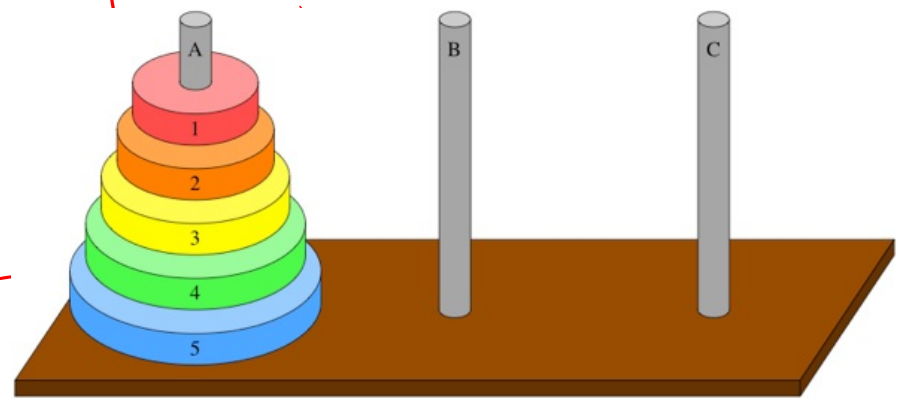
Tower of Hanoi

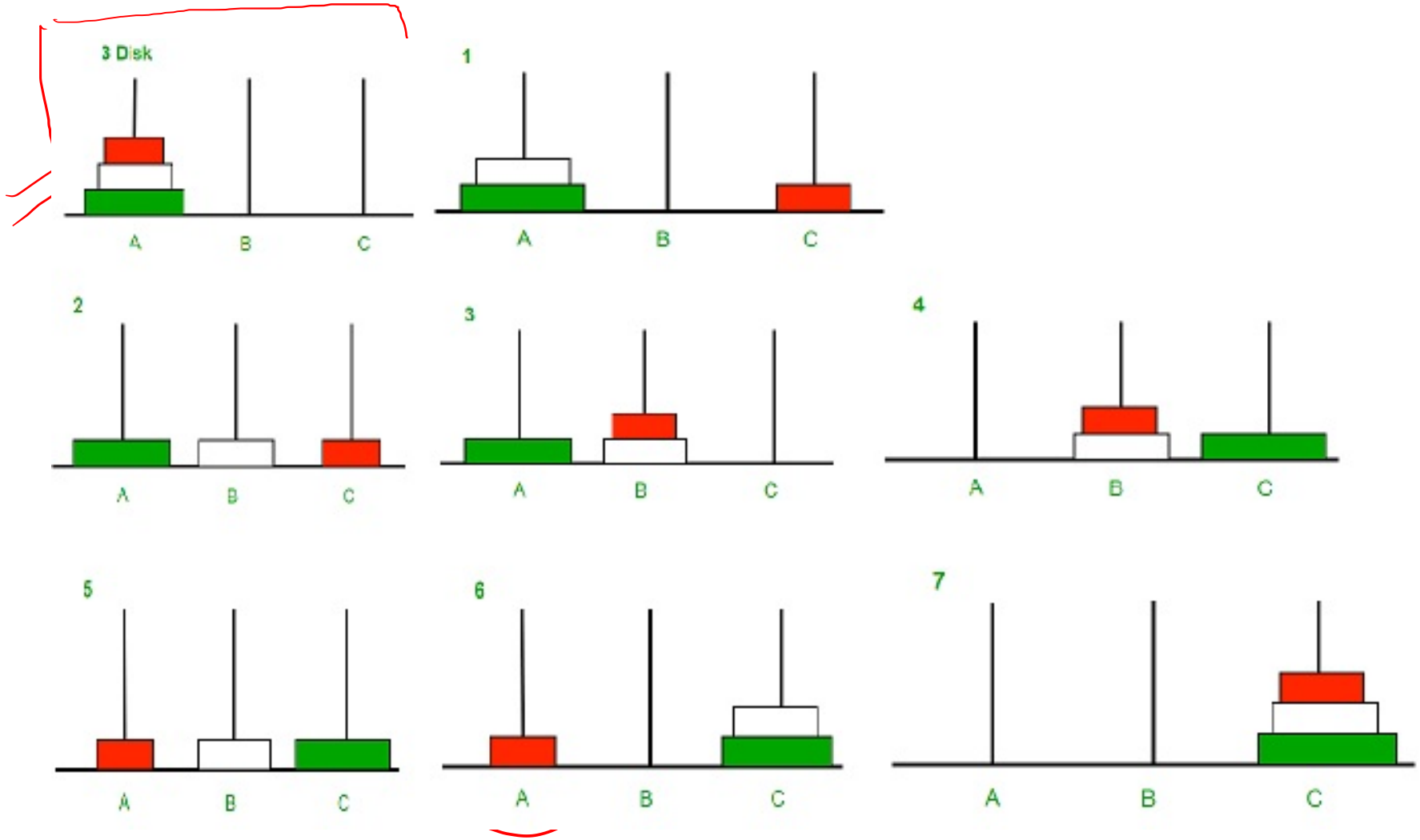
I/p:



Rules

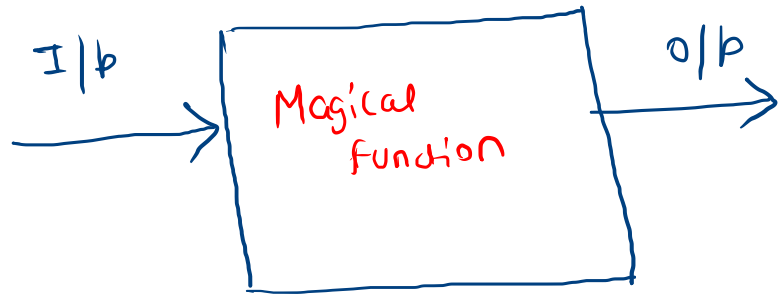
1. Only 1 disc can move at a time
2. In each move, pick the upper disc from one of the rod and place it on another rod
3. No Larger disc will placed on Smaller disc

Ví dụ:

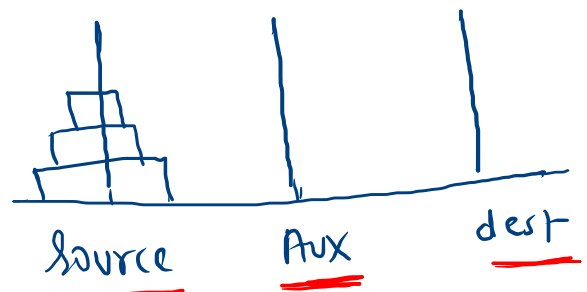


7

Recursive solution

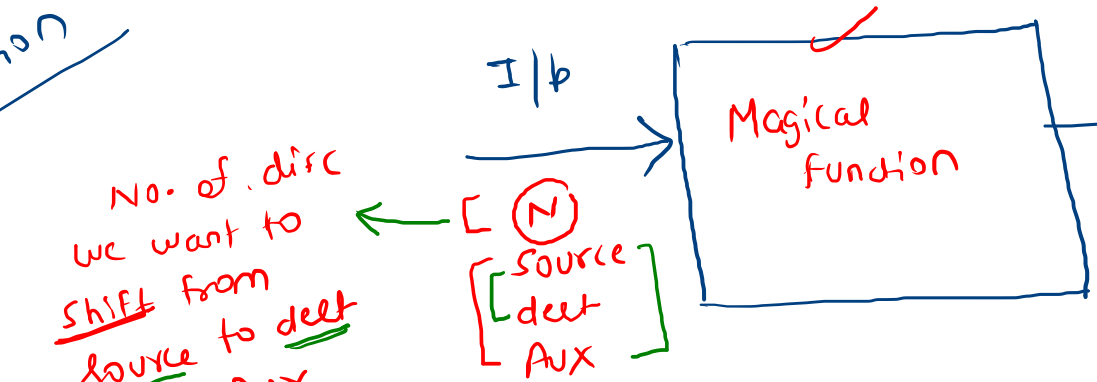


Logic

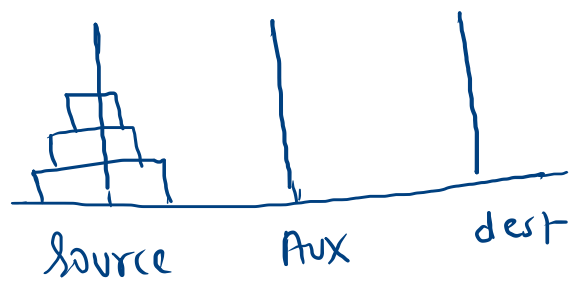


→ Ultimately, we want to shift all disc from source to dest. with help of Aux rod.

Recursive Solution

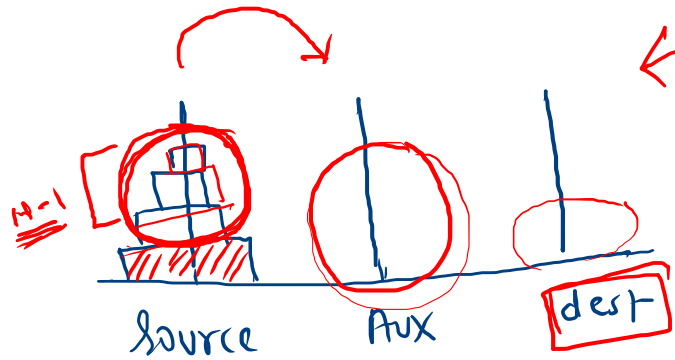


Count the No. of steps to move N disc

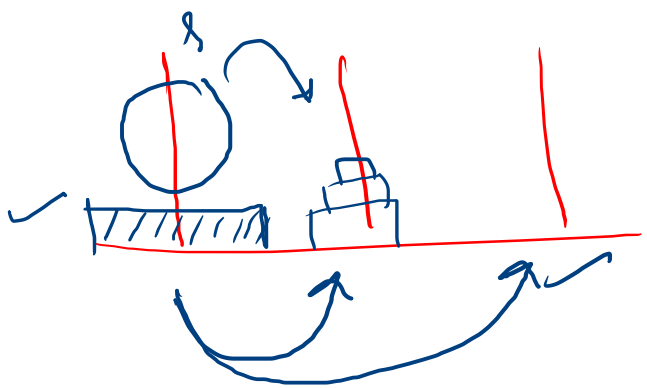


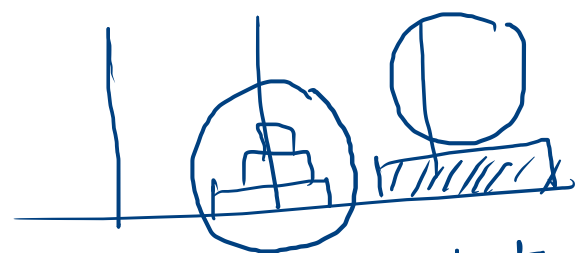
→ Ultimately, we want to shift all disc from source to dest. with help of Aux rod.

Recursive code



(N-1) disc → Sour → Source
dest → Aux
Aux → dest

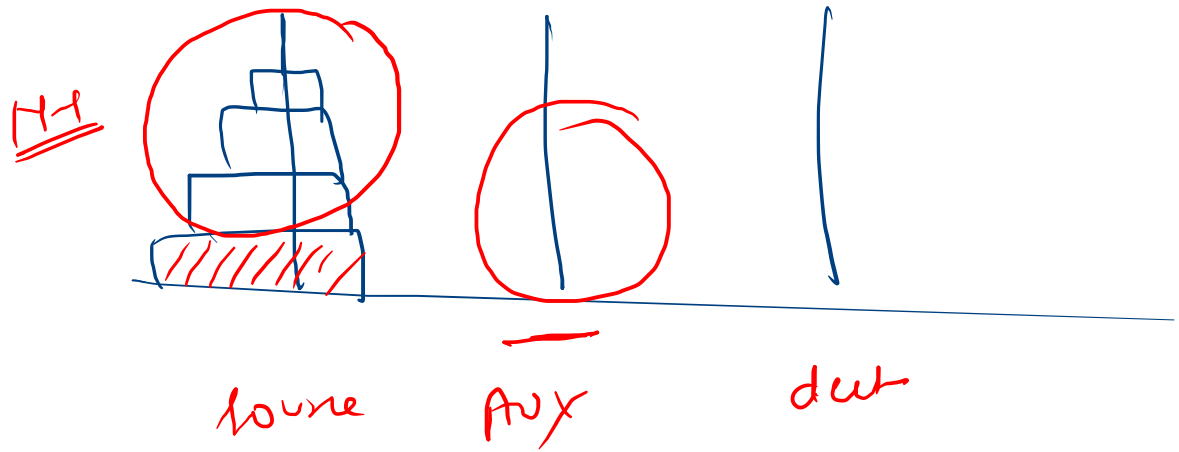




[Aux]

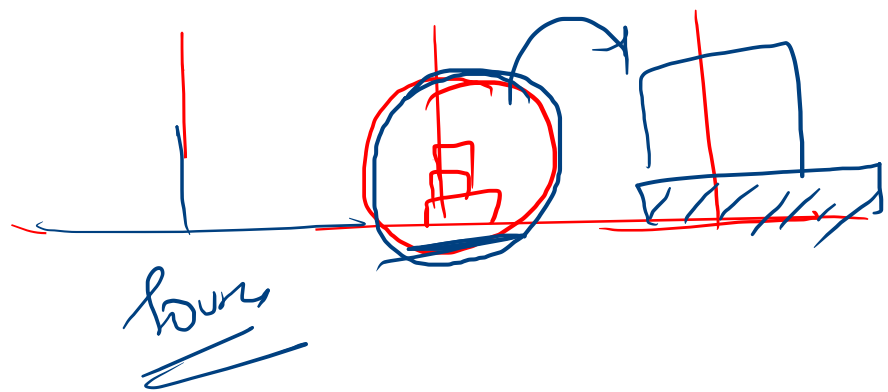
[Sour] [dest]

```
int Count = 0

void TOH ( int N, int Source, int dest, int Aux)
{
    // Base condition

    TOH (N-1, Source, Aux, dest)
    cout << " move disk (N) from rod Source to dest ";
    Count++;
    TOH (N-1, Aux, dest, Source);
}
```



ToH(N-1, source, AUX, dest)

Shift



ToH(N-1, AUX, dest, source)

Base condition

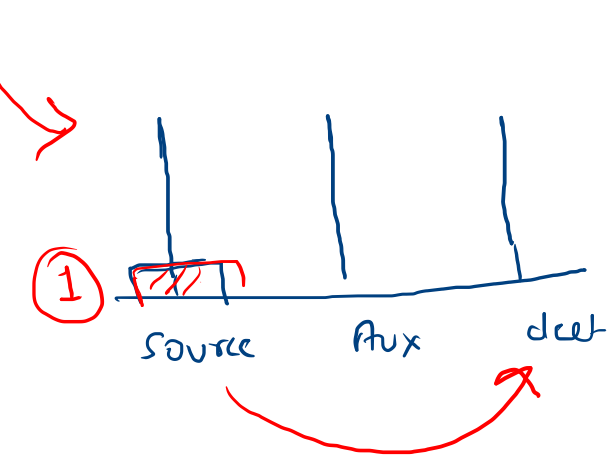
Smallest valid Input → check output

```
// void TOH ( int N, int source, int dest, int Aux)
```

N = 0, 1, 2, 3 . . . .

0 → Not Possible

1 → 2 → ...



Base Cond'

```
if ( N == 1 )
{
    cout << " move disk 1 from rod source to dest ";
    count ++ ;
    return ;
}
```

Recursive code

```
void TOH ( int N, int source, int dest, int Aux)
{
    if( n>0 )
    {
        TOH (N-1, source, Aux , dest )
        cout << " move disk (N) from rod source to dest ";
        count ++ ;
        TOH (N-1, Aux , dest , source.) ;
    }
}
```