



Your Ultimate Guide To Landing Top AI roles



DECODE
AiML

2.3.9

Tower of Hanoi

Problem: Tower of Hanoi Puzzle consists of 3 towers namely source, Auxiliary and Destination. you have n disks arranged in the source tower in such a way that smaller disk is kept over larger disks.

Goal: you need to move n disks from src tower to dest tower using Aux tower by below rules

- ① only 1 disk can be moved at a time
- ② No disks can be placed on top of smaller disks.

For $n=3$



```
def ToH(n, src, Target, aux):
```

Base Case

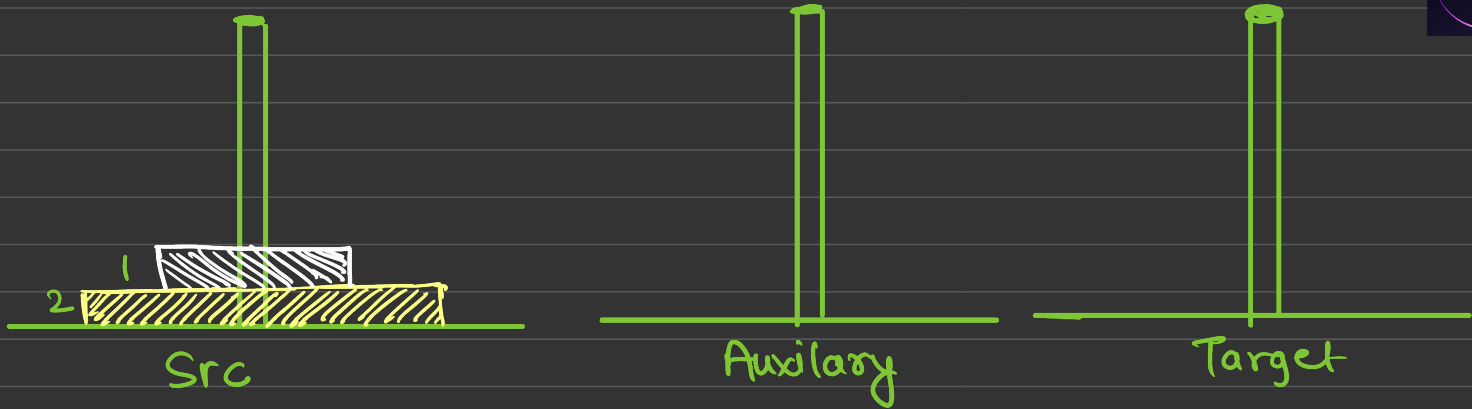
Recursive Case

Base Case



```
def TON(l, src, Target, Aux)  
    → Move l from Src to Target
```

Recursive Case



def ToH(2, src, Target, Aux)

- Move 1 from src to Aux \Rightarrow ToH(1, src, Aux, Target)
- Move 2 from src to Target
- Move 1 from Aux to Target \Rightarrow ToH(1, Aux, Target, src)

Recursive Case



def ToH(3, src, Target, Aux)

→ Move 1 from src to target

→ Move 2 from src to Aux

→ Move 1 from target to Aux

→ move 3 from src to target

→ move 1 from Aux to src

→ move 2 from Aux to target

→ move 1 from src to Target

ToH(2, src, Aux, Target)

ToH(2, Aux, Target, src)

```
def TOH(n, Src, Target, aux):
```

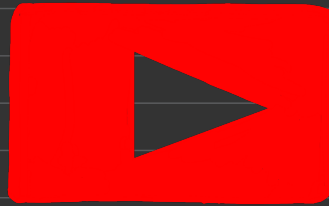
```
    if n == 1:
        print(f"Move 1 disk from {src} to {Target}");
```

Base
Case

```
    TOH(n-1, Src, Aux, Target)
    print(f"Move {n} disk from {src} to {target}")
    TOH(n-1, Aux, Target, Src)
```

Recursive
Case

Like



Subscribe