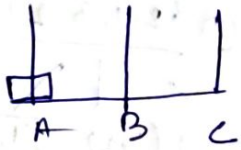
Tower of Hanoi

start, auxiliary, end

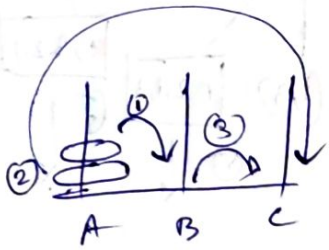
TOH (1, A, B, C)

→ Move disk from A to C using B



TOH (2, A, B, C)

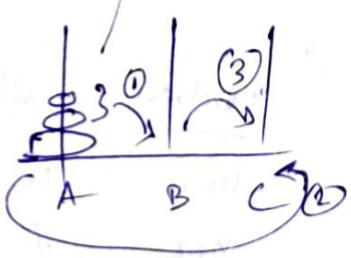
1. TOH (1, A, C, B)
2. Move disk from A to C
3. TOH (1, B, A, C)



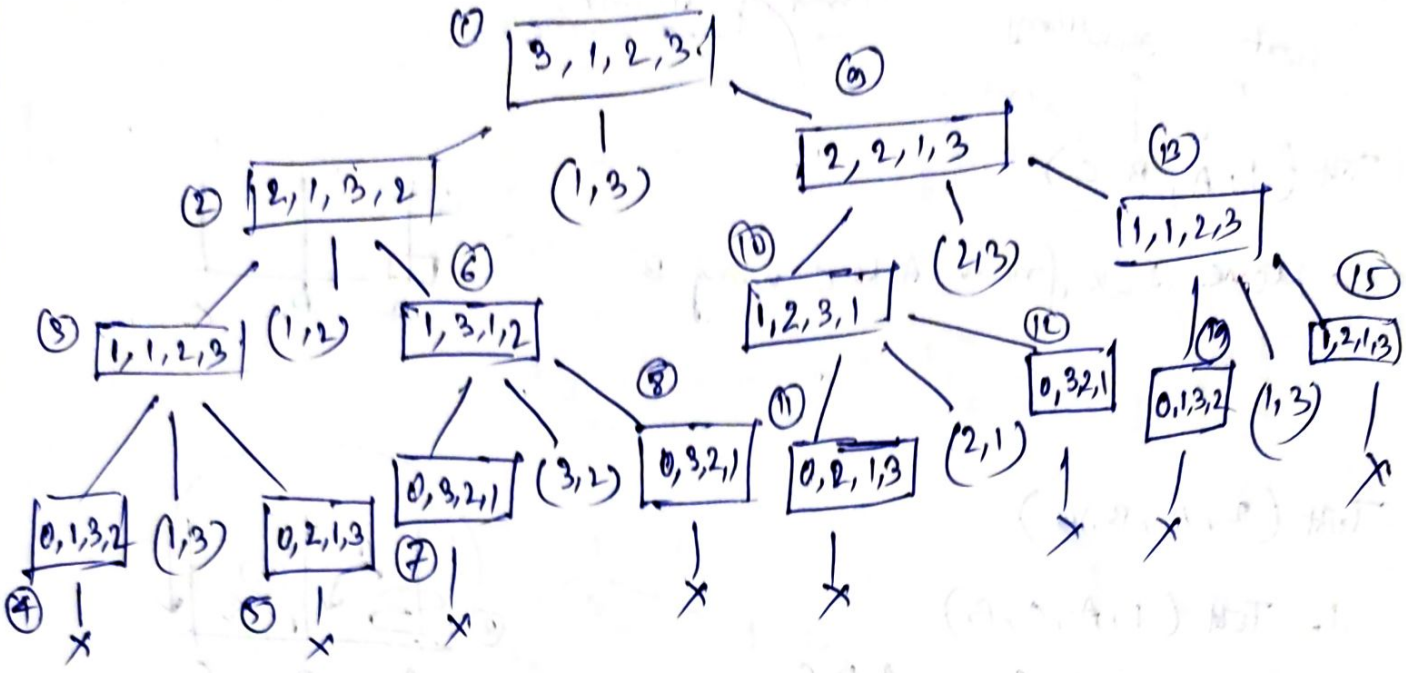
2 steps time ✗
• Using 2nd method

TOH (3, A, B, C)

1. TOH (2, A, C, B)
2. Move disk from A to C
3. TOH (2, B, A, C)




```
• void TOH (int n, int A, int B, int C)
  {
    if (n>0)
    {
      TOH (n-1, A, C, B);
      cout << A << " " << C;
      TOH (n-1, B, A, C);
    }
  }
```



output → (1,3) (1,2) (3,2) (1,3) (2,1) (2,3) (1,3)

Steps to follow

number of calls

no 3 → 15 = $1+2^1+2^2+2^3 = 2^4-1$

no 2 → 7 = $1+2^1+2^2+2^3 = 2^3-1$

So, TC = $O(2^n)$ ← (A)