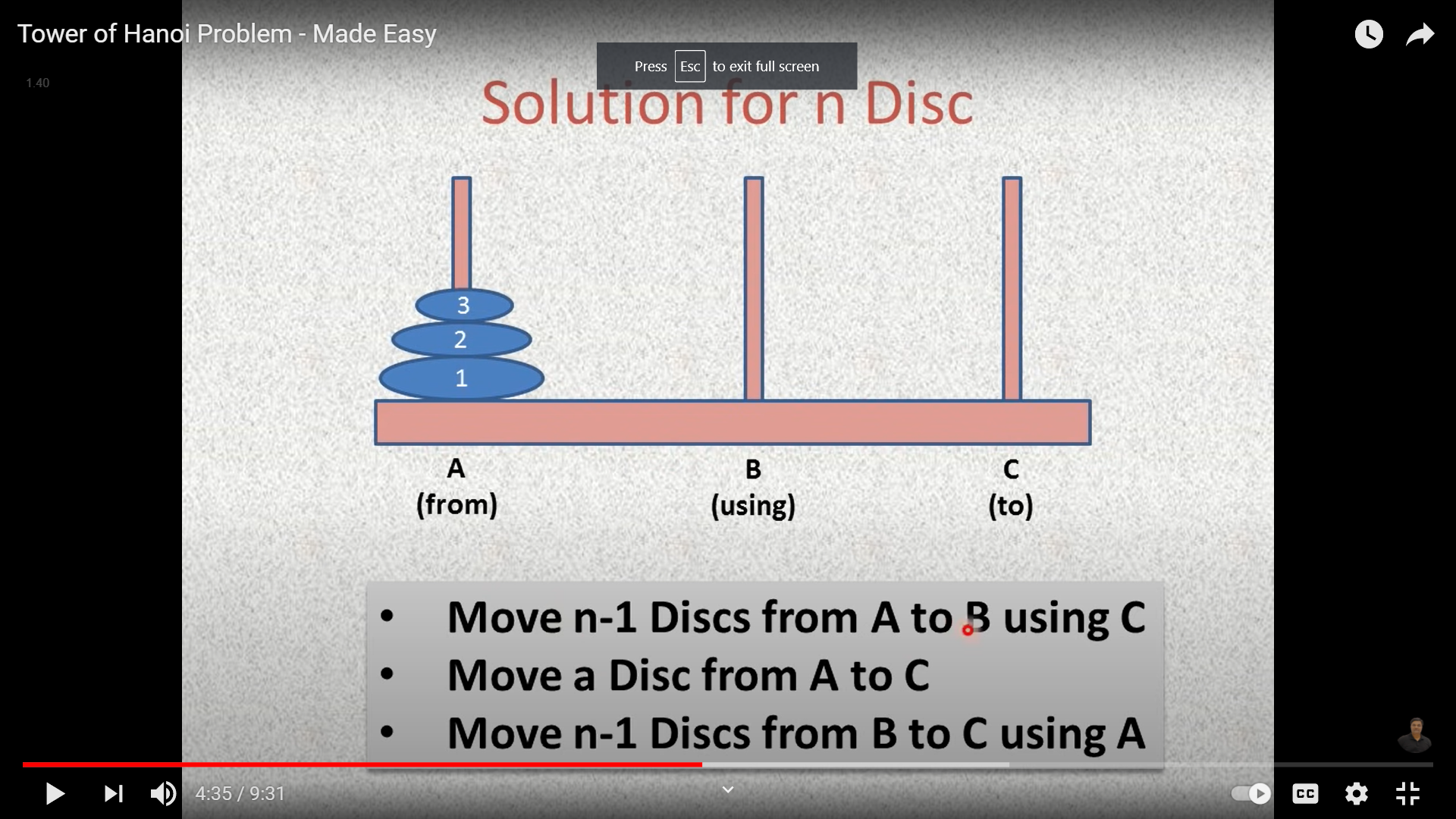
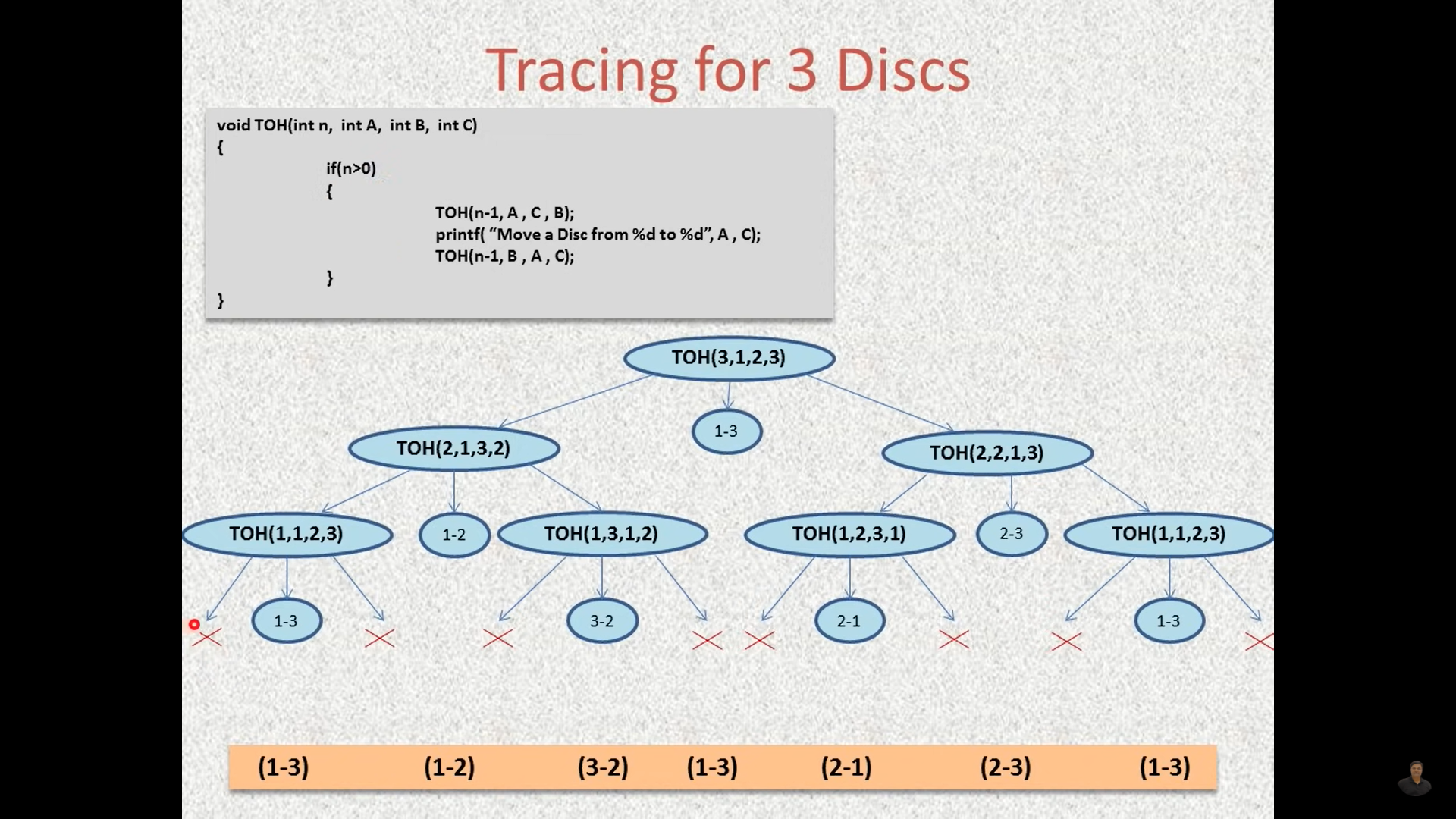
**Tower of Hanoi Recursive Algo:**





public class Main {  
 public static void TOH( int n , int A , int B, int C){  
  
 if(n>0){  
 *TOH*(n-1, A , C, B);  
 System.*out*.println("Move a Disc from tower " + A + " to tower " + C);  
 *TOH*(n-1, B , A, C);  
  
 }else{  
 return;  
 }  
 }  
  
  
 public static void main(String[] args) {  
  
 *TOH*(4, 1,2,3 );  
  
 }  
}

For n=1: 1

For =2: 3

For n=3: 7

For n=4: 15

For n=5: 31

**For general n: 2n -1**

Understanding Recursion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Tower A** | **Tower B** | **Tower C** | **Remarks** |
| **TOH(n, A, B, C)** |  |  |  | Goal to move the n disk from A to C via B |
| **TOH(n-1, A, C, B)** | 1 largest disk | n-1 other disks | 0 disks | Moved n-1 disk to tower B |
| **Print statement** | 0 disks | n-1 other disks | 1 largest disk | Moved largest disk to C |
| **TOH(n-1, B, C, A)** | n-1 other disks | 0 disks | 1 largest disk | Moved n-1 disk to tower A |