

Tower Of Hanoi

Given :- ① There are 3 towers. Tower 1 has n disks, where n is a positive number. Tower 2 and Tower 3 are empty.

② The disks are increasingly placed in terms of size such that the smallest disk is on the top and largest disk is at ~~bottom~~ bottom.

③ You are required to

③.1 Point instructions to move the disks

③.2 From tower 1 to tower 2 using Tower 3

③.3 Rules :-

⇒ move 1 disk at 1 time

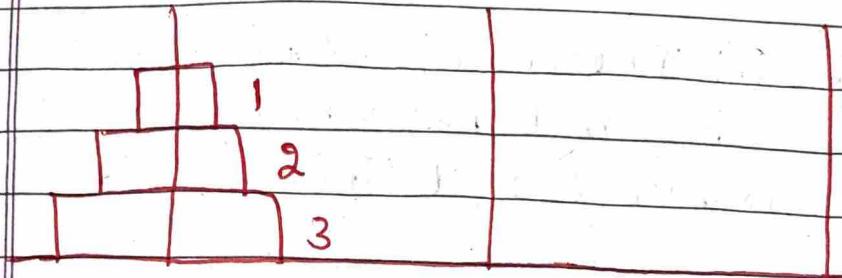
⇒ never place smaller disk under larger

⇒ you can only move the disk at the top

Input :- A number n , represents disks
 A number n_1 , represents id Tower 1
 A number n_2 , represents id Tower 2
 A number n_3 , represents id Tower 3

Output :- $n [n_1 \rightarrow n_2]$

Understand First



A

B

C



सबसे पहले हमारे पास 3 Tower हैं

lets suppose A, B & C. 3 तower

A पे 3 disk मर्ही है (3) सबसे बड़ी

बड़ी सबसे नीचे 3 में के ऊपर (2) जो (1)

से बड़ी है but (3) से choti है 1 के नकरे

ऊपर पड़ी है (1) disk वो सबसे छोटी है



अब हमको क्या करना पड़ता है?

⇒ हमको Point the instructions करना है That

are required to move all disks from

A to B, using C. 3 rule का दिया गया

22 अप्रैल 2023 का मेरा काम क्या था :-

- One at a time
- bigger disks can't be placed above smaller
- you can only move top most disk

* अब हम से क्या करना है high level thinking

जो code likhda henge then low level thinking से पता करता है कि code क्या करता है

High level Thinking

$toh(n, t_1, t_2, t_3)$

⇒ Expectations :- $toh(3, A, B, C)$

* हम इसे expect करते हैं $toh(3, A, B, C)$ जो

उस instructions point होगी जिसका tower

A में पड़ी हुई 3 disks tower B में चल

जारी tower C में use करने लगे

Faith :- $\text{tob}(2, A, B, C)$

(*)

3rd faith ye afलाई ये नहीं हो सकता है

2nd instructions peint दिए गए हैं from

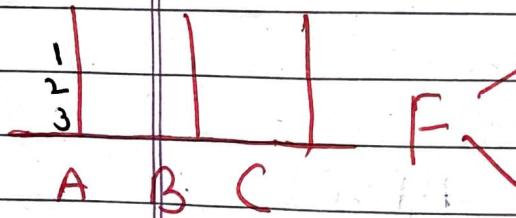
Tower A to Tower B using C

size 2 disks hongi and rules

follow rule 5(2) 2¹ 4¹ H 4² 2¹

→ (3 rules) of starting H 4¹ 2¹ 2¹ of

Expectation Meets Faith :-



$$F_1 = \text{tob}(2, A, C, B)$$

$$3(A, B)$$

$$F_2 = \text{tob}(2, C, B, A)$$

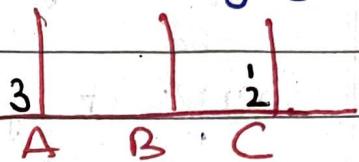
(*)

$\sum_{i=1}^{12} 4^i$ 4125 Faith के 2 294 ये i.e.

$\text{tob}(2, A, C, B)$ will peint valid
instructions to move 2 disks from

A to C using B and following 1-meules

* Agar इनकी सभी instructions की बात होती है कि disk(1) & disk(2) will shift from tower A to tower C using (B) & within rules.



* अब हमने कुछ से एक instruction likha i.e. 3[A,B] मतलब, जो disk(3) tower A me डाल दिया जाना है वही हमने tower (B) me डाल दी।

3[A, B] F2 = toh [C, B, A]

जो 2 disks के लिए instructions Point करती है वही हमने 2 disks within rules करती है।

2 disks (1) & (2) tower B pe pochuch

जारी करके from tower C using A

Par 3[A, B] मतलब कि instructions follow

की तरीके 2 disks (1) & (2) tower C

से 3rd के tower B chali jारी करकी using A.

Code :-

```

import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args)
        throws Exception {
    }
}

```

// write your code here

```

Scanner Scn = new Scanner(System.in);
int n = Scn.nextInt();
int t1id = Scn.nextInt();
int t2id = Scn.nextInt();
int t3id = Scn.nextInt();
Toh(n, t1id, t2id, t3id);
}

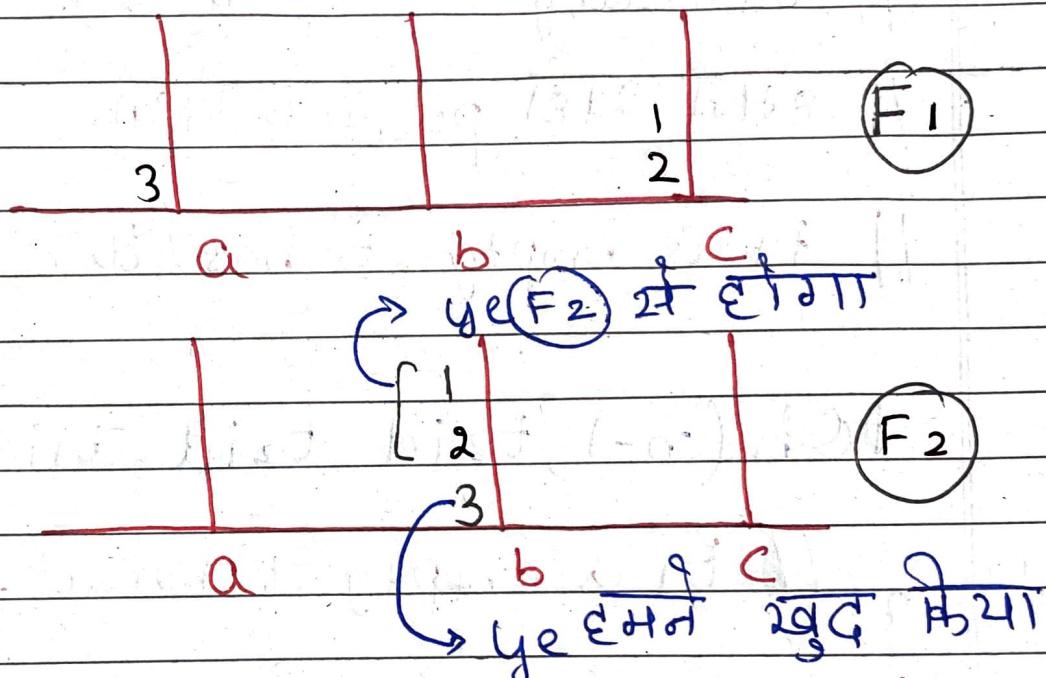
```

}

// E \Rightarrow Toh(3, a, b, c) \Rightarrow It is supposed to
 print valid (rule-following)
 (a/b/c) instructions to move 3 disks from
 a to b using c.

// Faith 1 \Rightarrow $T(n, a, b, c) \Rightarrow$ It will print
 (a) valid instructions to move 2 disks from
 a to c using b.

// Faith 2 \Rightarrow $T(n, c, b, a) \Rightarrow$ It will
 print (b) valid instructions to move
 2 disks from c to b using a.



* Done instructions follow करी न।
 3-पानी का म दोजारा।

public static void $T(n, int id_1, int id_2, int id_3)$

```

Base   if (n == 0) {
    |
    |
    |
    |
    return;
}
    |
    |
    |
    |
    [
    ]
  
```

$tob(n-1, t_1.id, t_3.id, t_2.id);$

// t_1 has only 1 disc, t_2 is empty, t_3 has two discs

`System.out.println($n + "[" + t_1.id + "-" + t_2.id + "]$)`

// basically इन से यह बाग फैला दिया जाएगा

// इसके लिए print $t_2.id$ करेंगे

// t_1 is empty, t_2 has 3rd disc, t_3 has 2 discs

$tob(n-1, t_3.id, t_2.id, t_1.id);$

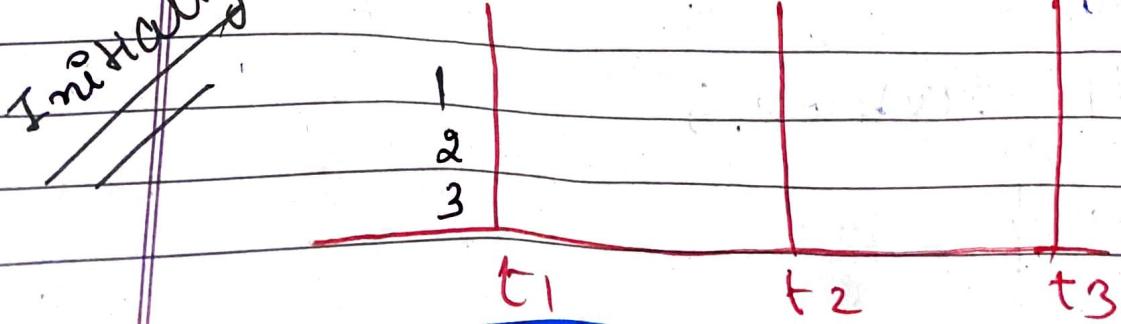
} // t_1 is empty, t_2 has all, t_3 empty

}

Low Level Thinking

Day Run

~~Initially~~



Public static void toh(int n, int tid, int t2id, int t3id){
if (n == 0) {

return;

}

toh(n-1, tid, t2id, t3id);

System.out.println(n + "[" + tid + " -> " + t2id + "]");

toh(n-1, t3id, t2id, tid);

(*) जैसा हम देख सकते हैं यह 2 calls

हैं लेकिन 'Pre area' में क्या है 'Post area'

मैं क्या है | 2121 काम 'in area' में होता है

(*) make a tree similar to zig-zag

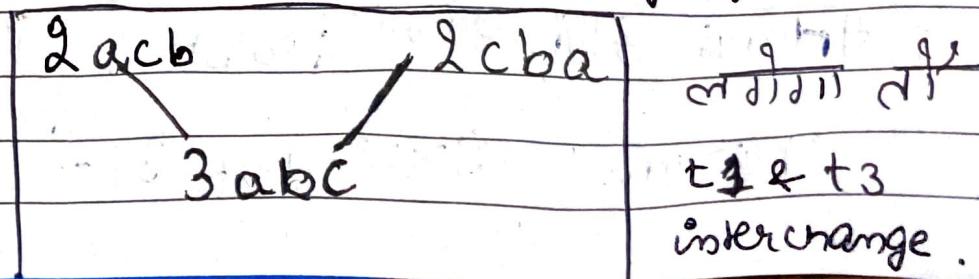
i.e. तीन discs हैं (a, b, c) नाम की

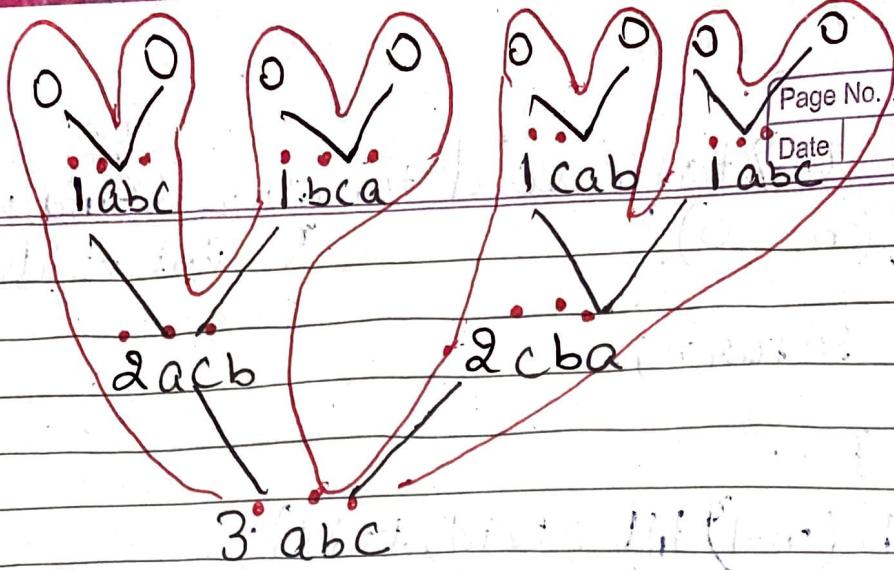
(*) हमें 'n' तक रखते हैं, tid me @ 31121,

t2id me (b) 31121 and t3id me (c).

(*) आप करने वाली ओर कि :- जब left side

लाल चलता है t2 & t3 interchange. जब 2nd cell





* Right call ५६८) - ०१२२) Change &

left call ५१२८) ०१०८) २ Change.

* Can use fingers to follow &

make stuff easier # beginners to the moon

* '0' pe serif return ५७०८) ५७०९) ना
matter नाही कृ३८८)

* first dot → ३abc ५३२९) ती call लागडी
५१२८) २acb पोहुच ०१२८) & lga इसपे १st dot,

first line चला १abc pe ०१२४) & lgea चला dot

३rd zero aaya return ०१२४)

* ३rd वापस ०१२४) १abc pe ०१२८) 2nd dot

महला अक दीजा Point So, [1 [a→b]] - ०

३rd लगा तीसरा dot ३२८) वापस call

लगा नीचे 'आरामी' & 'post area' के लिए
कुछ दूर होती है। जो यहाँ 'pre-area' के

नहीं होता था।



अब $2acb$ के लिए 2nd dot mtlb
point होगा & instruction $\text{मार्गी } SO, [a \rightarrow c]$



अब मार्गी वापस आये and 'pre'-के लिए

नहीं & लगाए 1st dot on $1bca$ then '0'

Ayega return $\text{मार्गी } 1bca$ के 2nd dot

mtlb instruction point $\text{मार्गी } SO, [b \rightarrow c]$

अब मार्गी आये dot & zero के लिए return

होते हैं $2acb$ के 3rd लगाए 3rd dot

mtlb call हो नीचे मार्गी $3abc$ पर।



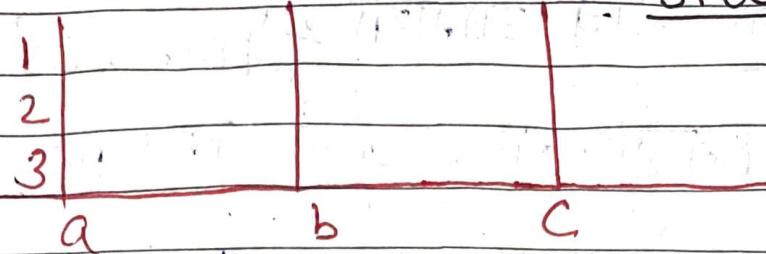
अब $3abc$ की 2nd dot mtlb instruction

होती है इसलिए हमने दीया ~~but~~ but

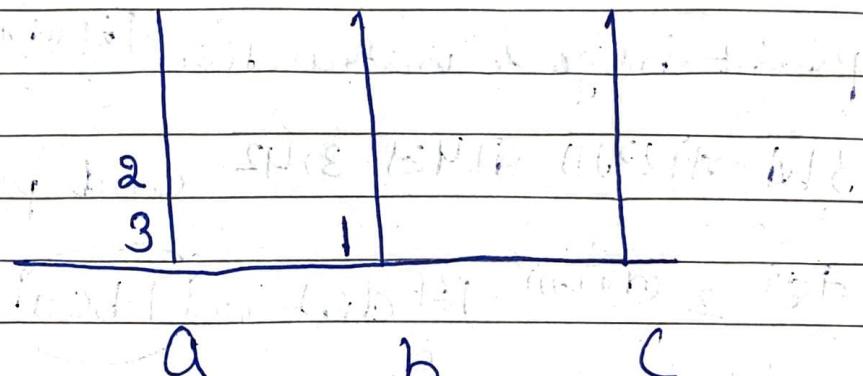
उससे पहले हमें left side is

complete SO, lets trace instructions

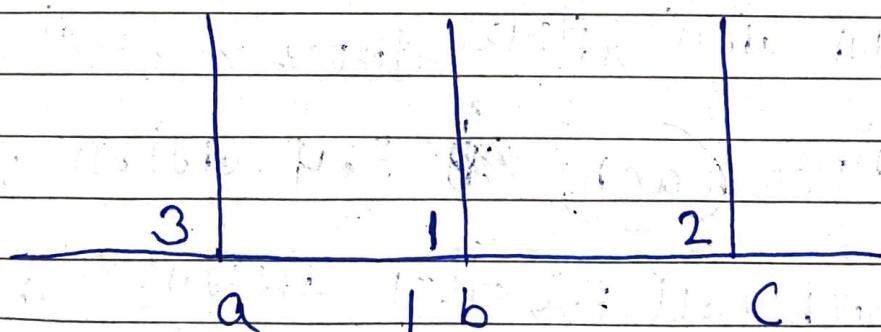
①, ② & ③

Visualization leftSide [F1]

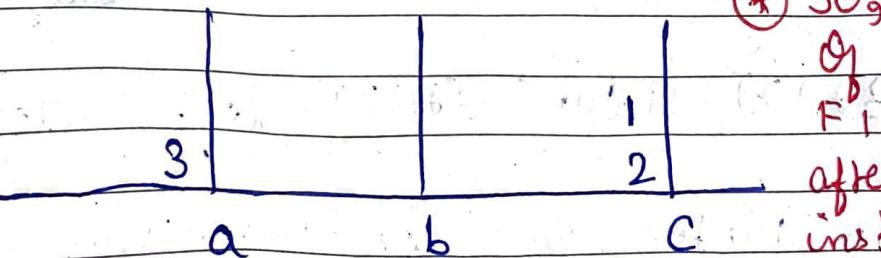
1 [a → b]



2 [a → c]



1 [b → c]

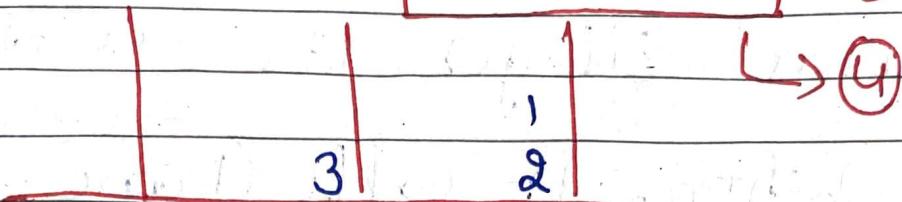


* So, At end
of left side
F1 is fulfilled
after following
instructions

* 3rd dot नीचे आएगा & 2nd dot mila (3abc)

कही मानलाए instruction 5th है i.e.

हम बोलेंगे SO, $3[a \rightarrow b]$ (यह दूसरी
बुक की है)



a b c

* right side of effect सिर्फ यह है
the end

Ab (3abc) pe 3rd dot because it is
giving its 2nd call. now (2cba) gets
its 1st dot which again calls (1cab)

which gets its 1st dot which calls gets 0

and returns but now (1cab) gets its
2nd dot meaning it will point

instruction SO, $1[c \rightarrow a]$ - 5

3rd milega 3rd dot '0' ke call then,
returns to (2cba)

* Now, acbba gets its 2nd dot

now, this means painting of 6
instruction will take place so, $2[\text{c} \rightarrow \text{b}]$

This will gets its 3rd dot

when, it calls 1abc

1abc gets its 1st dot which calls
'0' & return when, this happens

1abc will get its 2nd dot which
means painting takes place so, $1[\text{a} \rightarrow \text{b}]$ 7

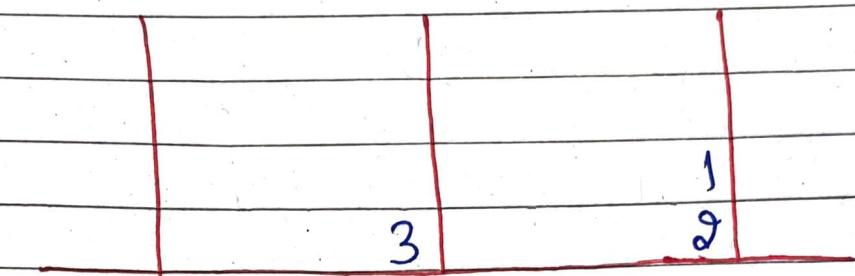
This means it gets third dot as
it calls zero & returns which
leads to an end of the program.

because there is no 'post area'
stuff happening.

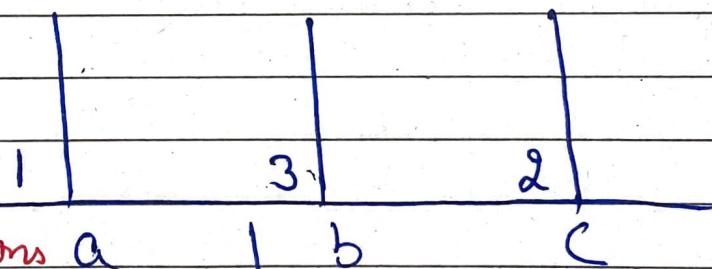
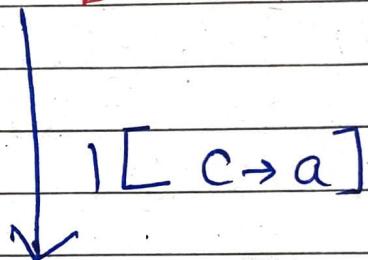
Visualization [F2]

(Right Side)

* using instruction ⑤, ⑥ & ⑦



a b c



* F2 instructions a
after fully being
followed lead us
to the very end
of F2 fulfilling promise to return the answer.

