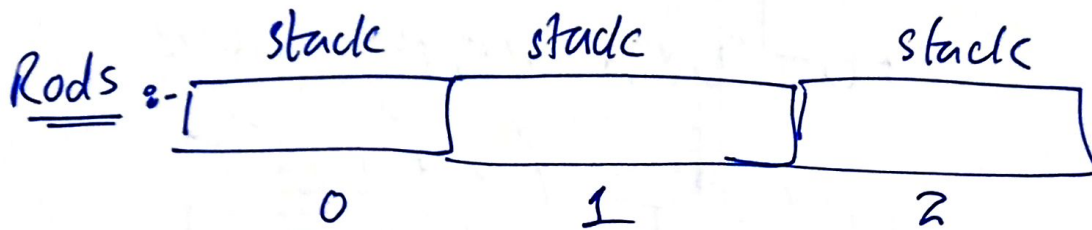


(1)

Two global variables:-

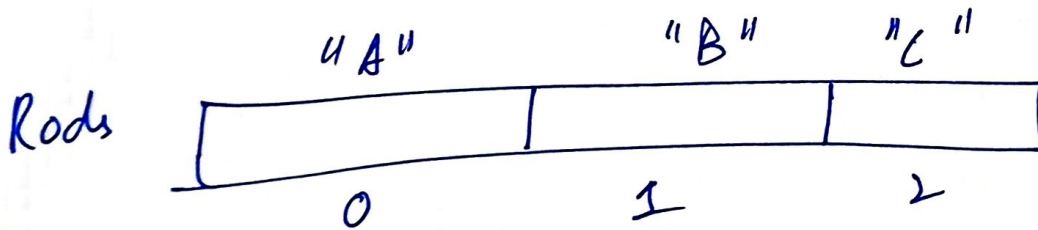
$\left\{ \begin{array}{l} \text{size} \leftarrow \text{steps} = 0 \\ \text{vector} \langle \text{stack} \rangle \text{ Rods}:- \end{array} \right\} \rightarrow \text{Line 27 \& 28 executed}$



main() {

Rods.pushBack (Stack (" A");
 " (" B");
 " (" C");

Three stacks pushed in



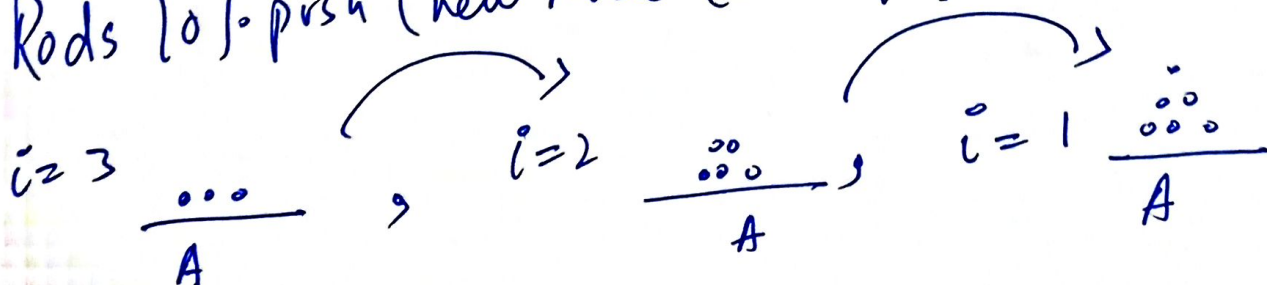
A & B & C stacks \rightarrow size = 0.

Get disk \rightarrow (3) \rightarrow we are taking 3 as disk.

lets populate "from" stack "A"

for (i = 3 ; 3 >= 1 ; i--)

Rods [0].push (new Node (Disk [i] 3))



tower of Hanoi (3 , ⁽²⁾ $\frac{0}{\text{from}}$, $\frac{2}{\text{to}}$, $\frac{1}{\text{aux}}$)

Now

ooo
ooo

A
size = 3
(from)

B
size = 0
(aux)

C
size = 0
(to)

step++ \rightarrow step = 1

(3 % 2 == 0) false.

for (3 (= 1 ; step++)

(pow(2, 3) - 1)

Entering the iterative loop, which will do the whole thing.

if (1 % 3 == 1) \rightarrow TRUE.

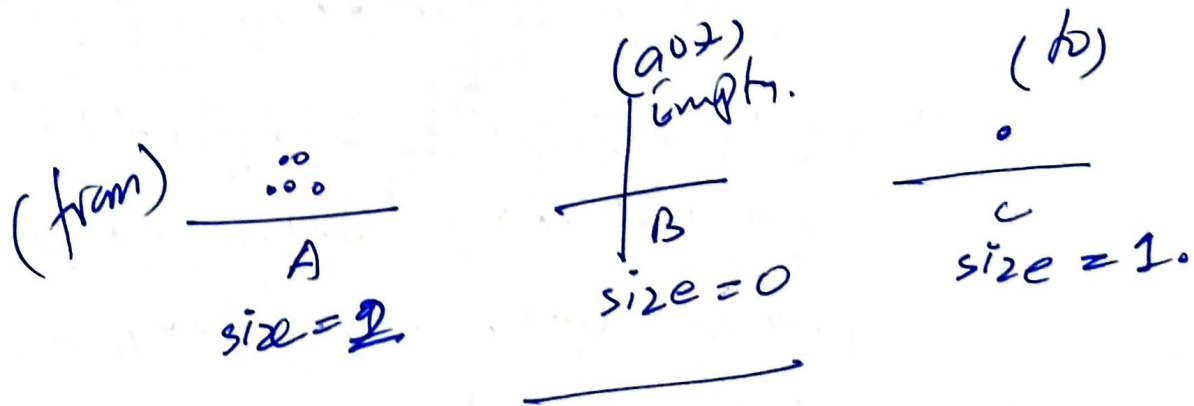
move disk(from, to) \rightarrow TRUE

if (Rod[to].getsize() == 0 || Rod[from].getsize() != 0)

size = 0 TRUE. size = 2.

actual Movement(from, to) \rightarrow TRUE.

this function just take the top 4 moved it to the to Rod. ⁽³⁾

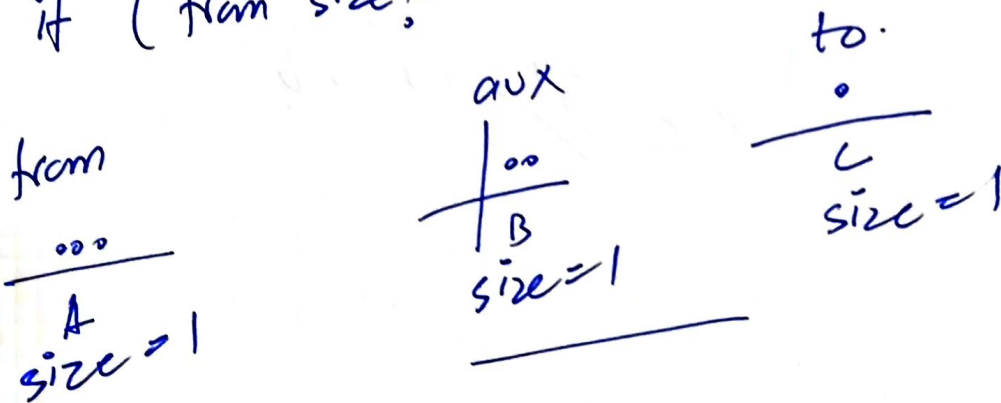


step 4 2:-

else if (2 % 3 == 2)
 moveDisk(from, aux) \rightarrow TRUE.

actual MoveDisk (from, to).

if (from size != 0 & to.size == 0).

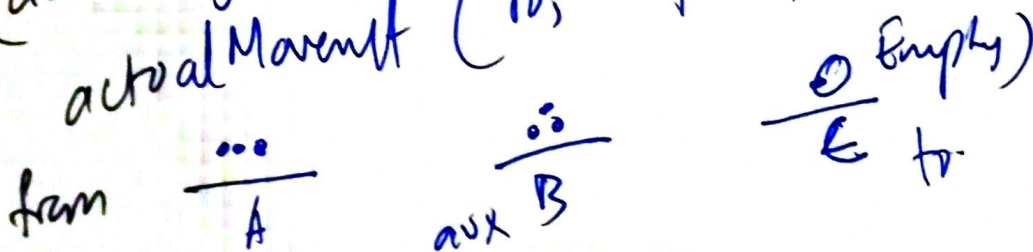


step 3

step 4 \rightarrow step = 3.

move(aux, to)

(aux length = 2 > to.length = 1) TRUE.
 actual MoveDisk (to, from).



④

step 4

step++ \rightarrow step = 4.

if ($4 \% 3 == 1$) TRUE.

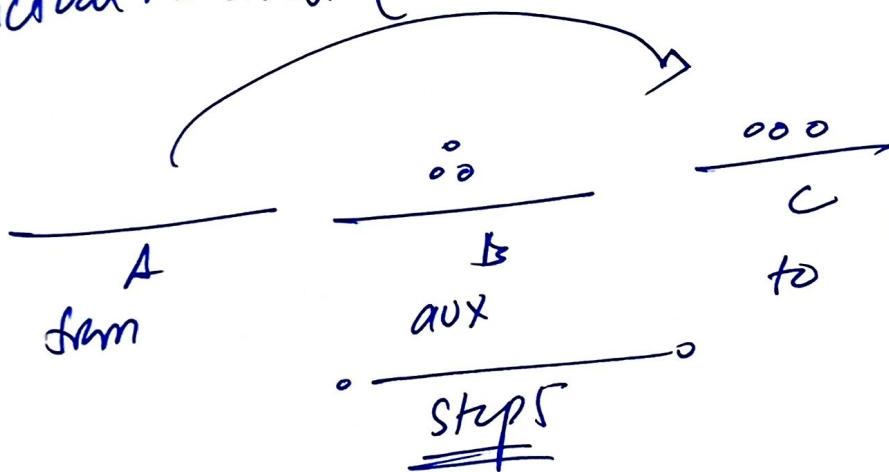
moveDisk (from, to) \rightarrow TRUE.

else if

(from.size != 0 & to.size == 0)

So, $A \rightarrow C$

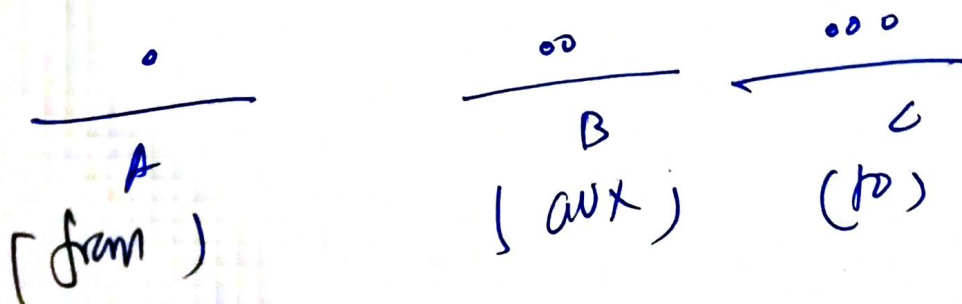
actual Movement (to, from)



else if ($5 \% 3 == 2$) TRUE.

$2 == 2$
moveDiskDisk (from, aux).

if (A.size == 0 & B.size != 0) TRUE.



3

step 6

step \rightarrow 6

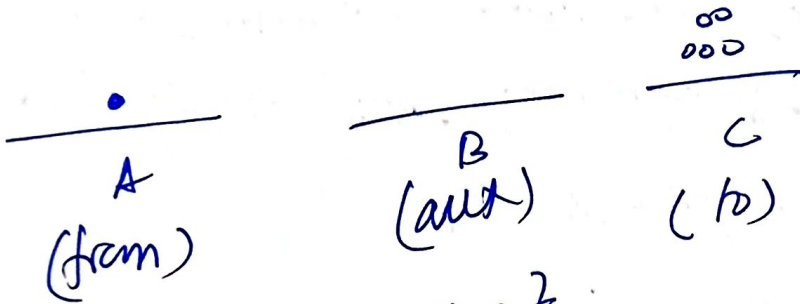
if (6 % 3 = 0)
0 = 0

moveDisk (aux, to)
(B) (C)

(59) line

else if ($\frac{\text{to size } 3}{(to)} > \frac{B \cdot \text{size } (2)}{(aux)}$) TRUE.

actual Movement (from, to) \rightarrow TRUE.



step 7

(step \rightarrow 7)

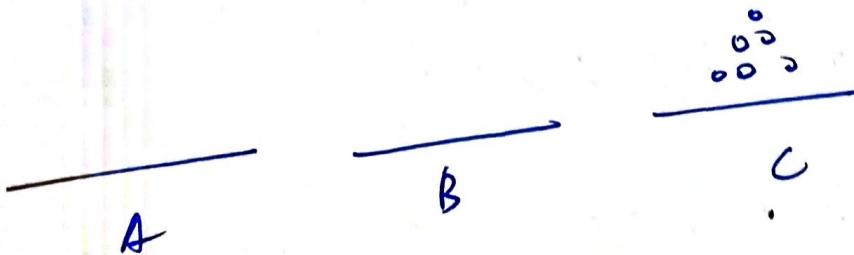
if (step % 3 = 1)

7 % 3 = 1
1 = 1

move Disk (from, to)

(60) line

else if ($\frac{\text{from size } (3)}{(from)} > \frac{\text{to size } (2)}{1}$)
actual movement (from, to)



return 0;