

DSA Notes

'lecture Of 20th October'

Agenda :- (Revision Introduction)

Ø → Point zig-zag - Cut

Ø → Tower of Hanoi - Cut

Q.

Print Zig-Zag

Given :- get following output for given input.

→ Input 1 → 1

Output → 1 1 1

→ Input 2 → 2

Output → 2 1 1 1 2 1 1 1 2

→ Input 3 → 3

Output 3 → 3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3

Input format :- A number n

Output format :- As discussed in given.

(*) ~~\$2101~~ * High level same E but ~~456~~ mainly

low level ~~H 3101~~ E ~~4211~~ 2 base

call ~~main~~ ~~101~~ ~~521012~~ ~~32101~~

~~456~~ analyse ~~as2d12~~

(*)

Input 1 \Rightarrow 1

Output 1 \Rightarrow 111

1 पाला Output
इसका कोई लिंग नहीं है।
Output नहीं

(*)

Input 2 \Rightarrow 2

Output 2 \Rightarrow ② 111 ② 111 ②

(*)

इसको देखा तो इसे पता चलता है।

कि यह ② likha है and अब +

पाला output है

(*)

Input 3 \Rightarrow 3

Output 3 \Rightarrow ③ ② 111 ② 111 ② 3 ② 111 ② 111 ② 3

(*)

Output 3 \rightarrow यह नहीं है क्योंकि phle '3' aaya

then ② wala output or we can say

that ③ मात्र for ② then '3' का output

यह ④ 2 ② का output complex है ③ का ④

(*) $1 = 111$

(*) $2 = \begin{matrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{matrix}$

(*) $3 = \begin{matrix} 2 & 1 & 1 & 2 & 1 & 1 & 2 \\ 3 & & 3 & & 3 & & 3 \end{matrix}$

High Level Thinking

(*) Expectations $\rightarrow \boxed{\begin{matrix} 2 & 1 & 1 & 2 & 1 & 1 & 2 \\ 3 & & 3 & & 3 & & 3 \end{matrix}} = P_{zz}(3)$

\Rightarrow माला की input 3 के लिए correct

Output 3 milega.

(*) Faith $\rightarrow \boxed{\begin{matrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{matrix}} = P_{zz}(2)$

\Rightarrow ऐसी faith 221 212 के input 2 के लिए
Correct output 2 ऐसी मिलेगा

(*) Expectation weds Faith $\rightarrow P_{zz}(3) = 3$
 $P_{zz}(2) = 3$
 $P_{zz}(2) = 3$

~~मानला यह कि वह प्रैन्ट [3] point करेगा।~~

(*)

~~Pzz Pzz(2) Then again [3] point again~~

~~Pzz(2) last तो Pzz(2) [3] point~~

(*)

lets express this in terms of 'n'

so, EWF \Rightarrow Pzz(n) = n || Point n

Pzz($n-1$) || Call($n-1$)

n || Point n

Pzz($n-1$) || Call($n-1$)

n || Point n

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();
```

```
pzz(n);
```

```
}
```

|| E \Rightarrow pzz(3) \Rightarrow 3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3

|| F \Rightarrow pzz(2) \Rightarrow 2 1 1 1 2 1 1 1 2

|| EWF \Rightarrow pzz(3) \Rightarrow 3 pzz(2) 3 p(zz2) 3

```
public static void pzz(int n) {
```

```
    if (n == 0) {
```

```
        return;
```

```
}
```

```
System.out.print(n + " ");
```

```
pzz(n - 1);
```

```
System.out.print(n + " ");
```

```
pzz(n - 1);
```

```
System.out.print(n + " ");
```

]} || we will find base case using
Dry Run

Dry Run

(*) public static void Pzz(int n) { if(n==0){ return; }

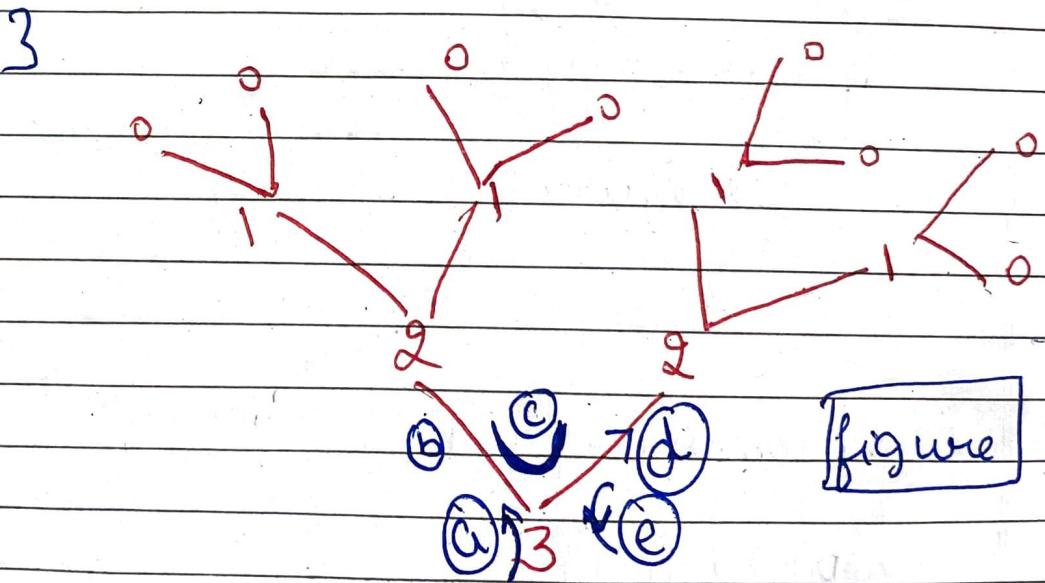
(a) System.out.print(n+" "); || Pre area

⇒ (b) Pzz(n-1); || Call back 1 (left call)

(c) System.out.print(n+" "); || In area

⇒ (d) Pzz(n-1); || Call back 2 (Right call)

(e) System.out.print(n+" "); || Post area



(*) ~~3~~ At ~~3~~ ~~2~~ ~~1~~ ~~Pzz(3)~~ ~~Y~~ and this question has 2 call backs so, for `Pzz(3)` it is going to call `Pzz(2)` twice.

(*) ~~3~~ ~~2~~ ~~1~~ ~~Pzz(2)~~ same 2 callback ~~at 1 2 3 4 5~~

To `Pzz(1)`:

* For $P_{zz}(1)$ 2 callbacks ~~match~~
to $P_{zz}(0)$.

* So, As shown ~~at 2nd~~ figure ~~at 2nd~~

* Line (a) is pre-area i.e. just before
the call i.e. just before 1st call

* Line (b) & line (d) are calls i.e.
call area & line (b) is 1st call & (d) is 2nd call

* Line (c) is 'in area' i.e. b/w the
two calls

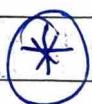
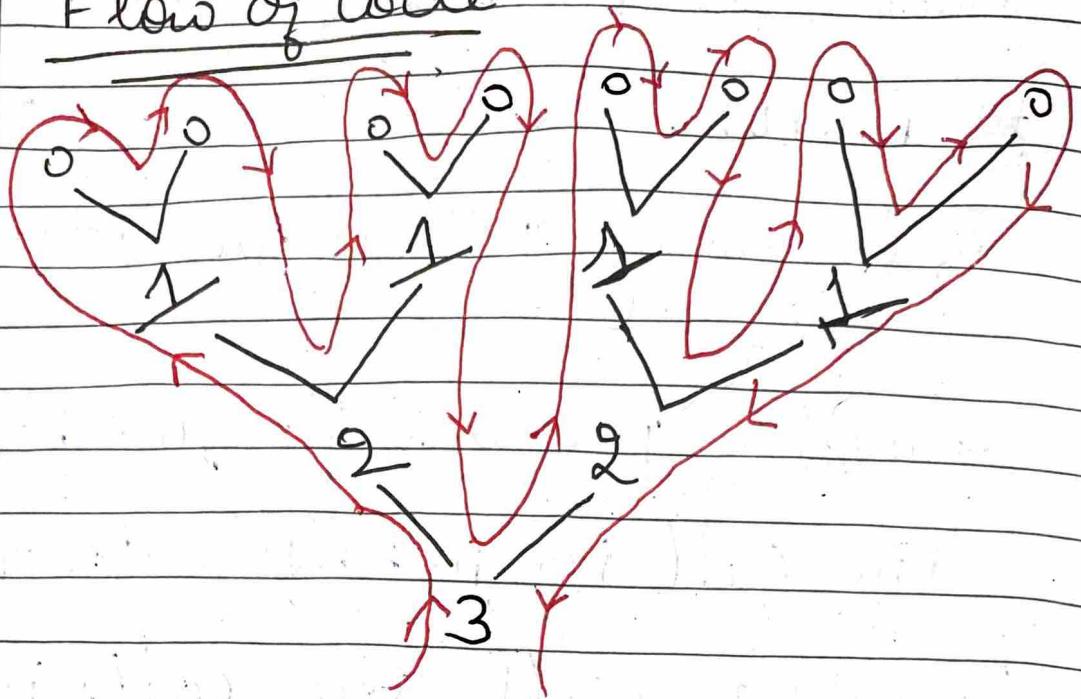
* Line (e) is 'post-area' just after
the 2nd call or we can say that
after both calls are done.

Steps

* phle 2nd HST or f^{or} E^t 2 calls $\frac{1}{2}$

* fir as shown in figure ; ye diagram
bna lo

Flow of Code



* Red line indicates flow of code



पहले मात्र (3) pe and (3) point हो जाए

then upरे मात्र हो and (2) point हो जाए

1 तक 2 (1) आया हो (1) point हो जाए then

(0) के कुछ नहीं हैं इसके return करता है

तो कुछ नहीं होता हो ये तो पर्वती

अब आजाए 'in area' से गति point हो जाए

जो कि (1) point होगा और अब it goes from

0 to 1 again मात्र (1) point हो जाए

'post area' के बाहर से 3142 (2) का 'in area'

3121 NT (1) ② print एल्गोरिदम का वापस (1) का

3142 NT (1) ① print एल्गोरिदम का 3121 (1) 3121

तो वापस 121 का return एल्गोरिदम का वापस

3121 (1) On the left side 3121 use

'Pre area' से ① Print एल्गोरिदम का already

return ab 2nd (1) का 312 'in area'

3121 +b वापस ① Print एल्गोरिदम का se

return के 2nd (1) का 'post area' से

too ① Print होगा

return एल्गोरिदम, then left wale (2) का

'post area' 3121 का ② Print होगा then,

(3) का 312 'in area' aagya 3121 (3)

Print एल्गोरिदम

* 3121 का एल्गोरिदम 421 421 Print का 31?

(Only for left side)

3 2 1 1 1 2 1 1 1 2 3

(*)

3-4 right side $\frac{1}{2} \text{ m}^2$ (2) $\frac{1}{2} \text{ m}^2$

'pre area' aaya ∇ (2) Point hoga fir

right side 1st (1) $\frac{1}{2} \text{ m}^2$ 'pre area'

aaya too (1) Point $\frac{1}{2} \text{ m}^2$, (0) 3-1($\frac{1}{2} \text{ m}^2$)

too self return $\frac{1}{2} \text{ m}^2$ then,

(1) $\frac{1}{2} \text{ m}^2$ st right side $\frac{1}{2} \text{ m}^2$ 1st ∇

∇ 'in area' 3-1(2) ∇ (1) Point hoga

3-4 (0) 3-1(2) ∇ return then,

1st ∇ right side (1) $\frac{1}{2} \text{ m}^2$ (post area) aya

too (1) Point hoga

(*)

3-4 (2) $\frac{1}{2} \text{ m}^2$ 'in area' aya ∇ (2) Point

$\frac{1}{2} \text{ m}^2$

(*)

3-0 3-4(2) 3-4(2) gye ∇ right side $\frac{1}{2} \text{ m}^2$ 2nd (1)

$\frac{1}{2} \text{ m}^2$ (pre area) too (1) Point $\frac{1}{2} \text{ m}^2$,

(0) aya ∇ return (1) $\frac{1}{2} \text{ m}^2$ 'in area' aya too

(1) Point hoga waps (0) return then,

right wale 2nd ① $\frac{1}{\cancel{1+2}}$ post area

so, ① return hoga i.e. ① Paint hoga

now, comes post area's ② on the

right side too ② Paint $\frac{1}{\cancel{1+2}}$ finally

we reach back to ③ $\cancel{1+2} \cancel{1+2}$

Paint hoga

④

For right side area $\cancel{1+1} + \cancel{2+1}$ Paint
 $\cancel{1+1}$?

2 1 1 1 2 1 1 1 2 3

⑤

3+12 finally 3+12 left side and
right side का 4+2 का 2 का रहा:

3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3

left side

right side

⑥

3+12 में पता है कि ⑥ से 3+12 का return करवा
कर उसको base case Snake code करलगा।

Q.

By Sir for Chat :-

public static void pzz(int n) {

if (n == 0) {

return ;

3

System.out.println(n + " Pre");

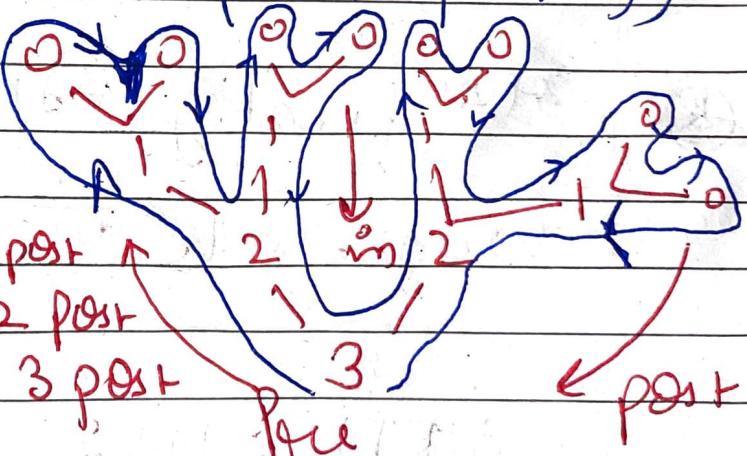
pzz(n - 1);

System.out.println(n + " in");

pzz(n - 1);

System.out.println(n + " Post");

3



RMD

3 Pre 2 Post 1 Post
2 Pre 3 in 2 Post1 Pre 2 Pre 3 Post
1 in 1 Pre

1 Post

2 in

1 Pre

1 in

1 Post

1 Post

2 in

1 Pre

1 in

* Tip for beginners

Use your finger

To follow Path