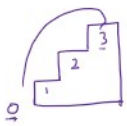
11-Dec-2021

15 December 2021 20:59

Print paths of stairs :- n = no. of stairs // 3



1 to 3

- 1 1 1
- 1 2
- 2 1
- 3

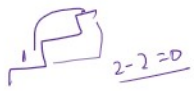
n = 4

- 1 1 1 1
- 1 1 2
- 1 2 1
- 1 3
- 2 1 1
- 2 2
- 3 1

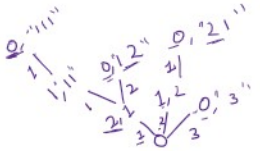
min. Jump = 1
to
max Jump = 3

n = 3

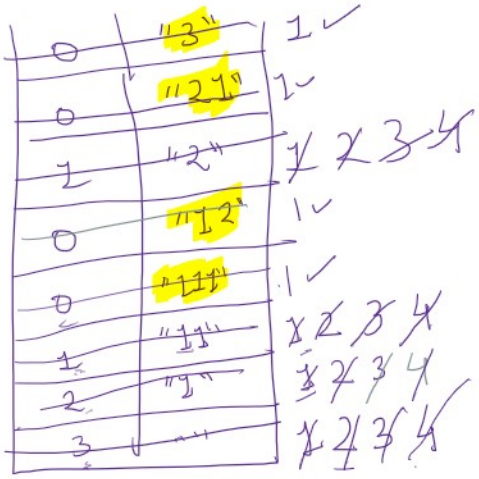
```
psp(n, res) {
    if (n == 0) { sout(res); return; }
    if (n-1 >= 0) {
        psp(n-1, res+"1");
    }
    if (n-2 >= 0) {
        psp(n-2, res+"2");
    }
    if (n-3 >= 0) {
        psp(n-3, res+"3");
    }
}
```



2-2=0



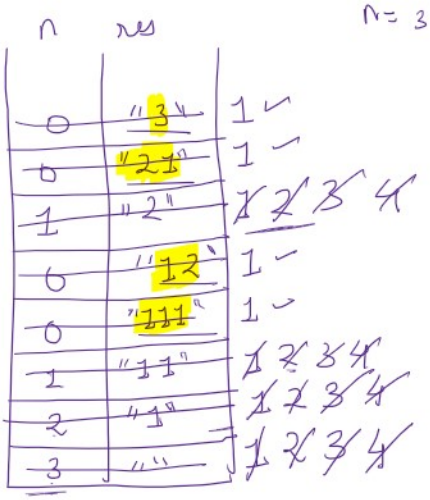
n res



n = 3

```
public static void printStairPath(int n, String res) {
    if (n == 0) {
        System.out.println(res);
        return;
    }
    if (n - 1 >= 0) {
        printStairPath( n: n - 1, res: res + "1");
    }
    if (n - 2 >= 0) {
        printStairPath( n: n - 2, res: res + "2");
    }
    if (n - 3 >= 0) {
        printStairPath( n: n - 3, res: res + "3");
    }
}
```

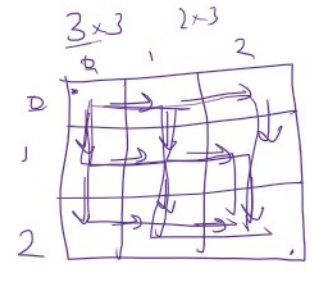
n res

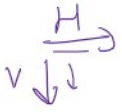


n = 3

```
public static void printStairPath(int n, String res) {
    if (n == 0) {
        System.out.println(res);
        return;
    }
    if (n - 1 >= 0) {
        printStairPath( n: n - 1, res: res + "1");
    }
    if (n - 2 >= 0) {
        printStairPath( n: n - 2, res: res + "2");
    }
    if (n - 3 >= 0) {
        printStairPath( n: n - 3, res: res + "3");
    }
}
```

Print Maze Path :-





HH VV

H V H V

H V V H

V H H V

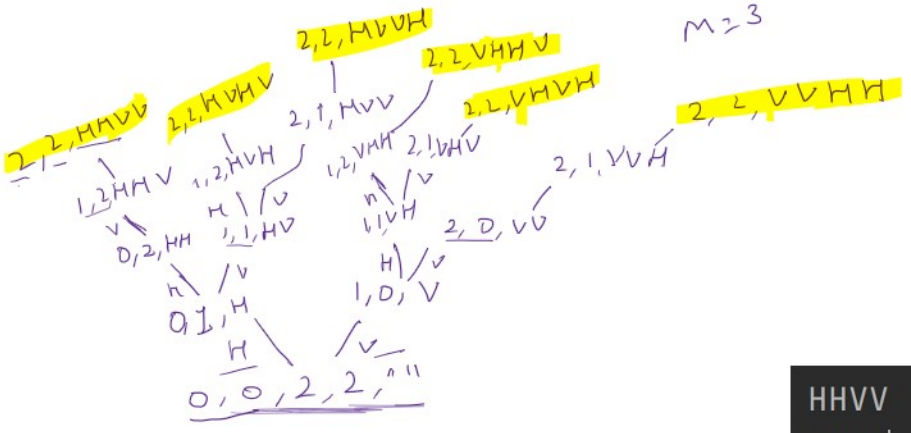
V H V H

V V H H

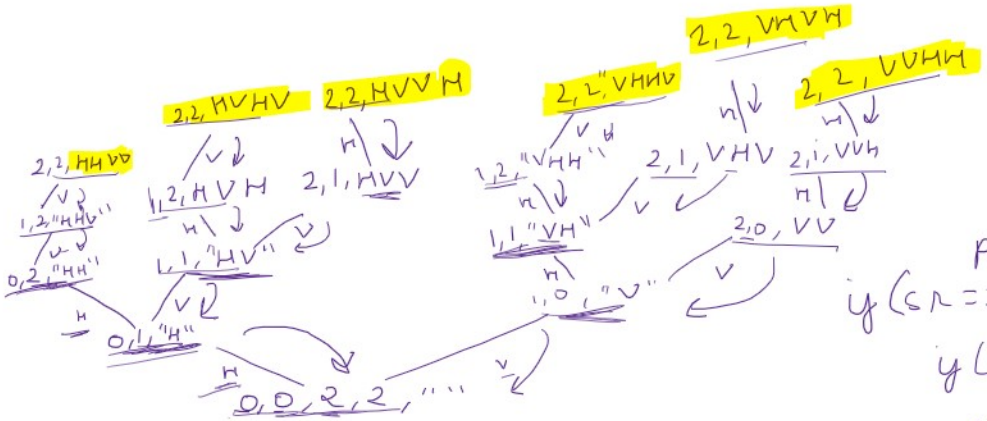
PMP(sr, sc, dr, dc, res)

n=3

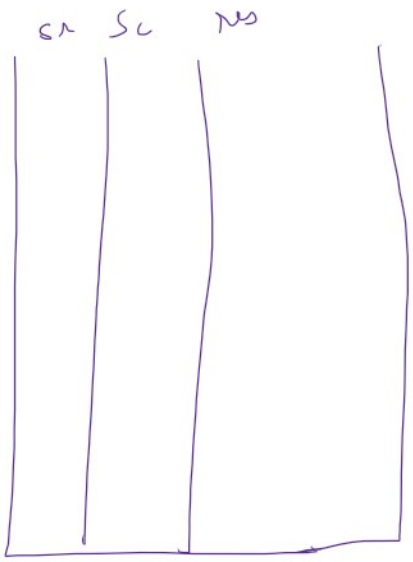
m=3



HHVV
HVHV
HVVH
VHHV
VHVH
VVHH




```
PMP (sr, sc, dr, dc, res){
if(sr==dr && sc==dc) sout(res);
if(sc+1<=dc){
PMP(sr, sc+1, dr, dc, res+"H");
}
if(sr+1<=dr){
PMP(sr+1, sc, dr, dc, res+"V");
}
}
```



```java
public static void printMazePath(int sr, int sc, int dr, int dc, String res) {
    if (sr == dr && sc == dc) {
        System.out.println(res);
        return;
    }
    if (sc + 1 <= dc) {
        printMazePath(sr, sc: sc + 1, dr, dc, res: res + "H");
    }
    if (sr + 1 <= dr) {
        printMazePath(sr: sr + 1, sc, dr, dc, res: res + "V");
    }
}
```

dr = 2

dc = 2

3 × 3

Output

H H V V

H V H V

H V V H

V H H V

V H V H

V V H H

H.W => Same question with diagonal Possibility.