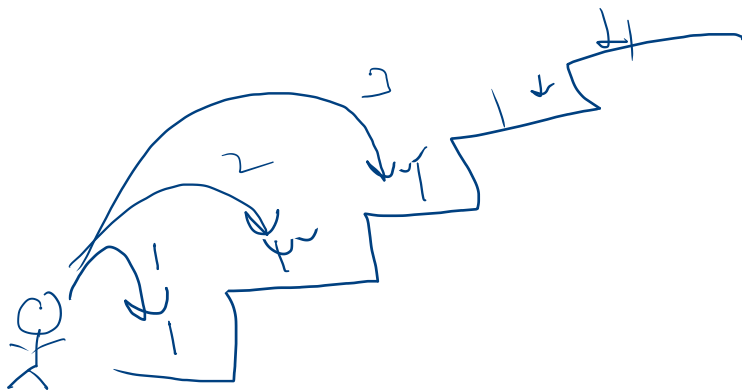
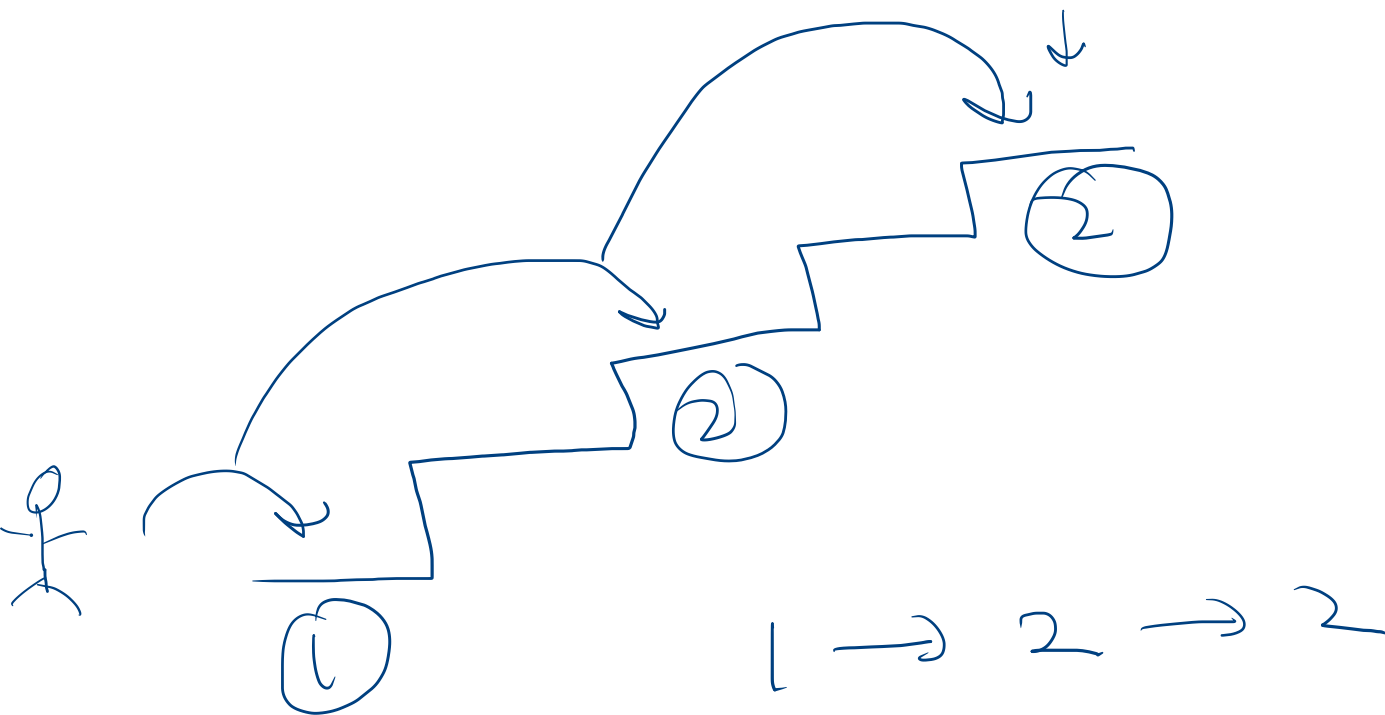


$n=5$





$$n=3$$



1 1 1

1 2

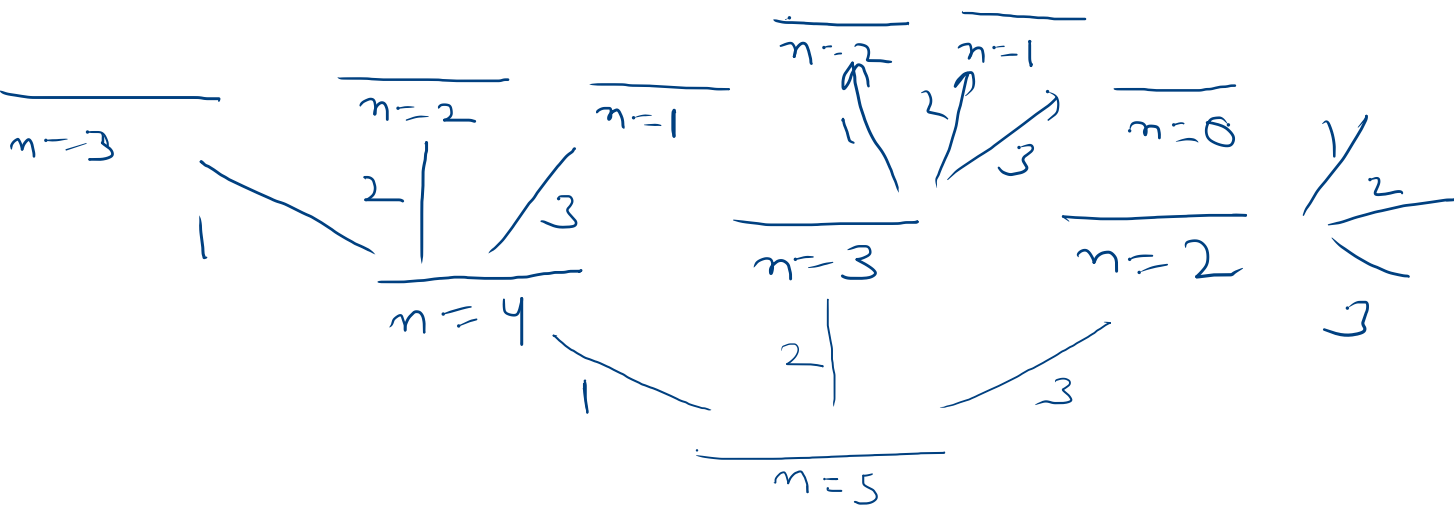
2 1

3

Source \longrightarrow destination

||

destination \longrightarrow Source



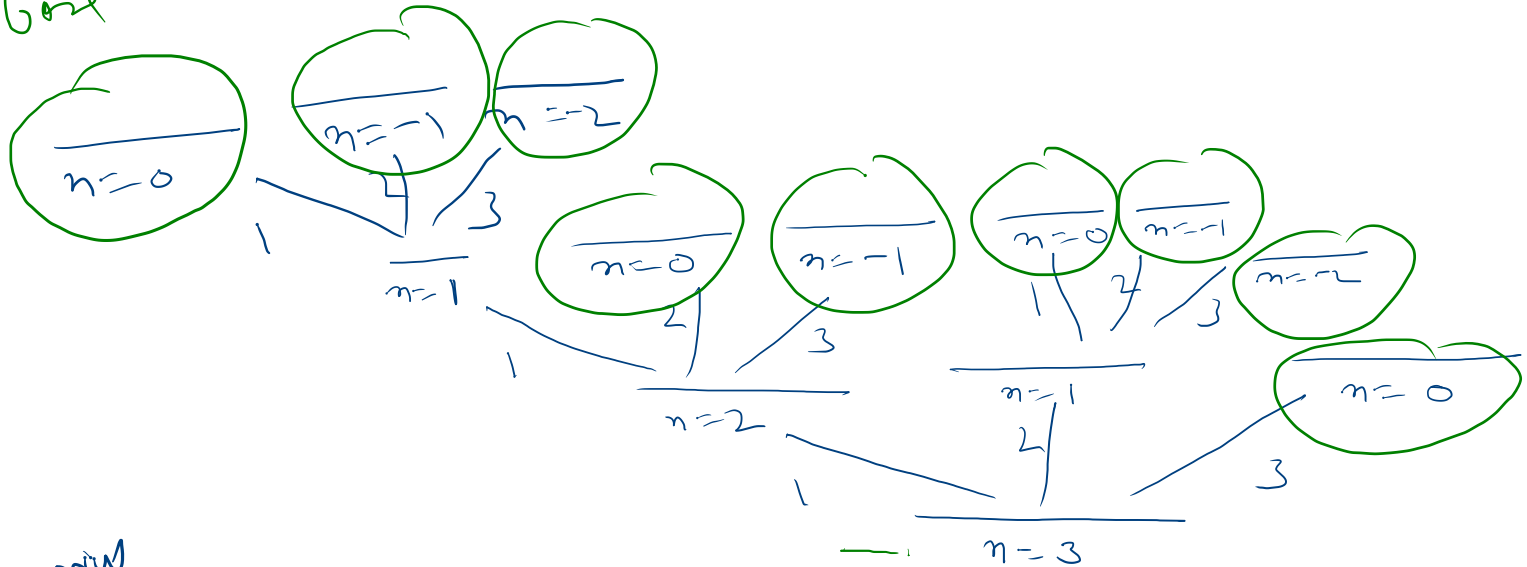
↑ small
Big

Big \rightarrow Small

base case₁ \Rightarrow $n=0 \rightarrow$ record ans

base case₂ \Rightarrow $n < 0 \rightarrow$ ignore

base

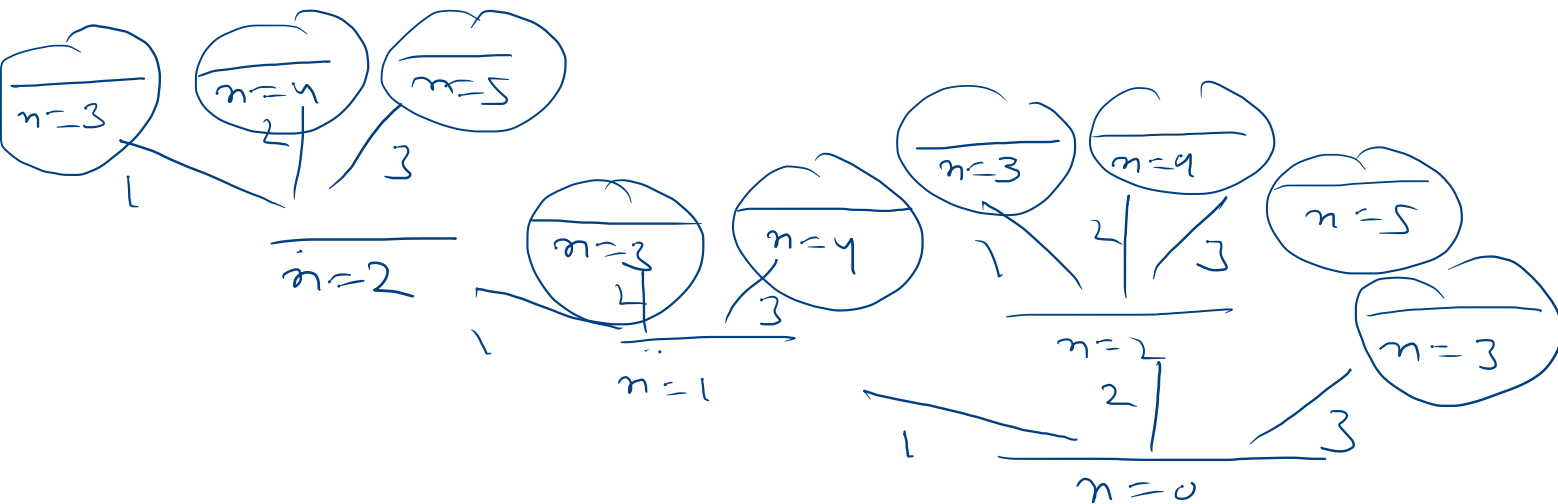


recursive
tree

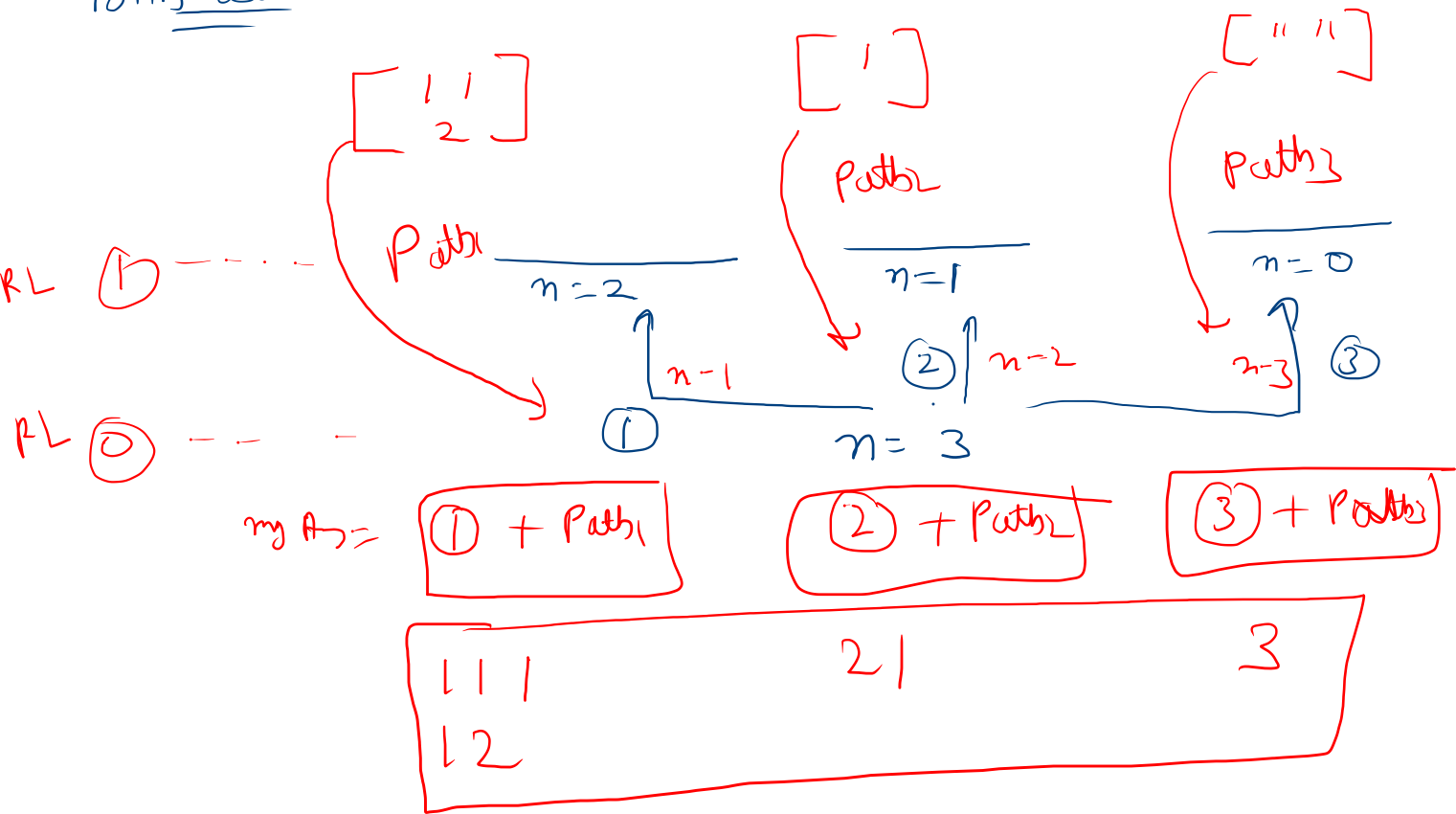
Small \rightarrow Big
 $\infty \rightarrow n=3$

base case $\Rightarrow n=3 \rightarrow$ second ans

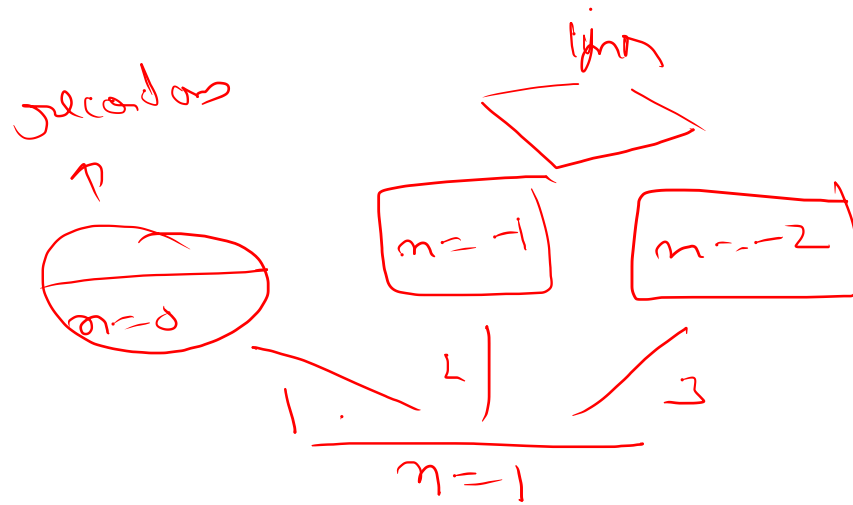
base case $\Rightarrow n > 3 \rightarrow$ ignore



Fairly code



base



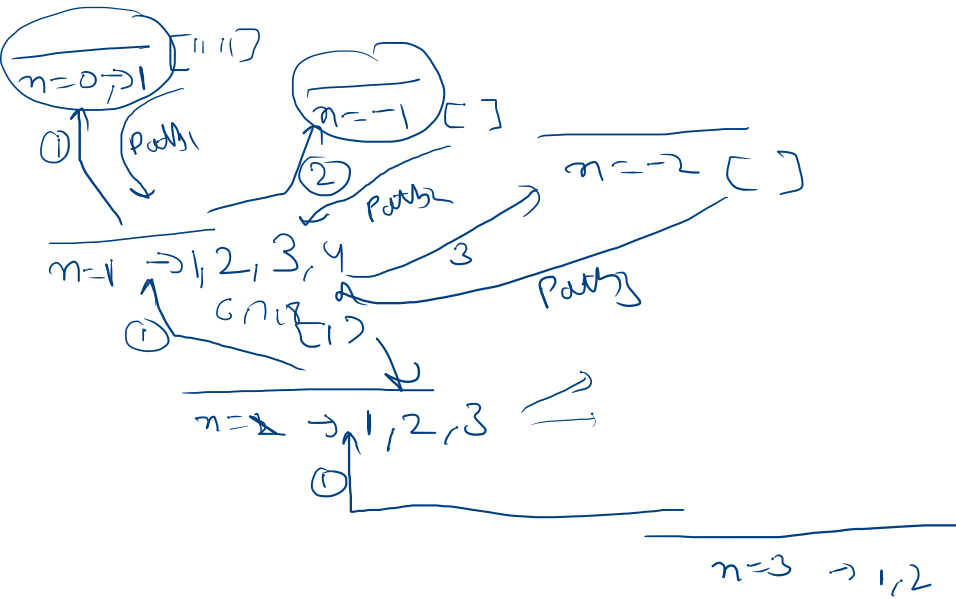
Smaller

Time $(n) \rightarrow$

$n \rightarrow 0$

3^n

Space \Rightarrow Recursive
Stack \vdash Output
Space
 (n)



```

public static List<String> solution(int n) {
    // base case
    if (n <= 0) {
        List<String> bans = new ArrayList<>();
        if (n == 0) {
            bans.add("");
        }
        return bans;
    }

    List<String> path1 = solution(n - 1);
    List<String> path2 = solution(n - 2);
    List<String> path3 = solution(n - 3);

    List<String> myAns = new ArrayList<>();

    // path1
    for (String str: path1) {
        myAns.add(1 + str + "");
    }

    // path2
    for (String str: path2) {
        myAns.add(2 + str + "");
    }
}

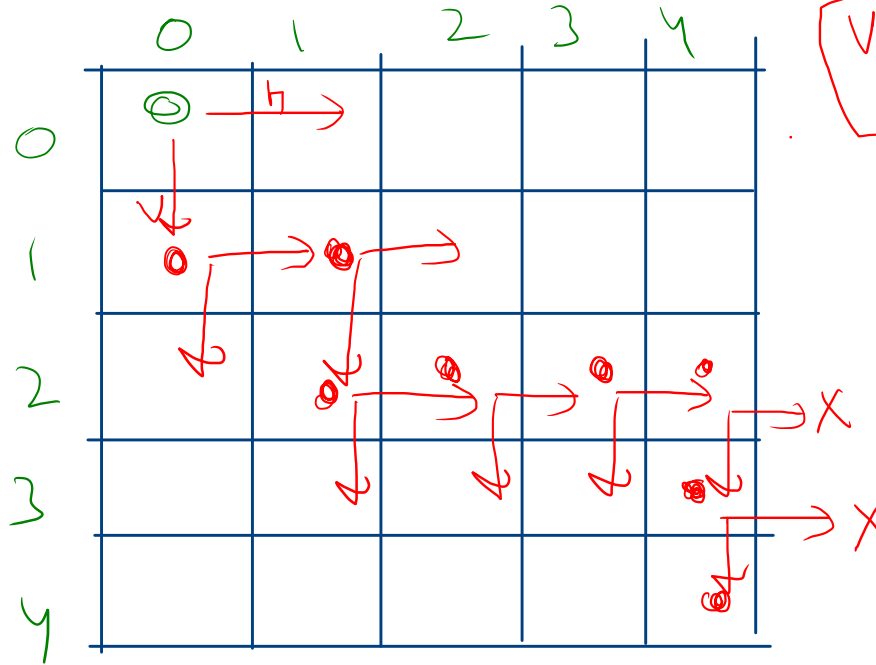
```

in general

k

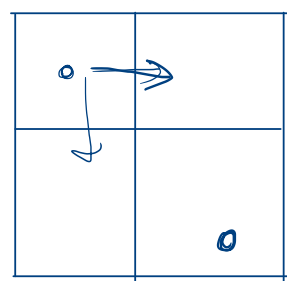
K^m

Get Maze Paths (Day 25)



src 0 1

$n=2$
 $m=2$



$(n-1, m-1)$
dest

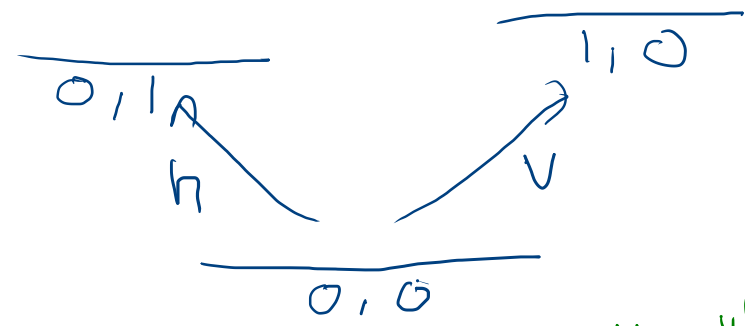
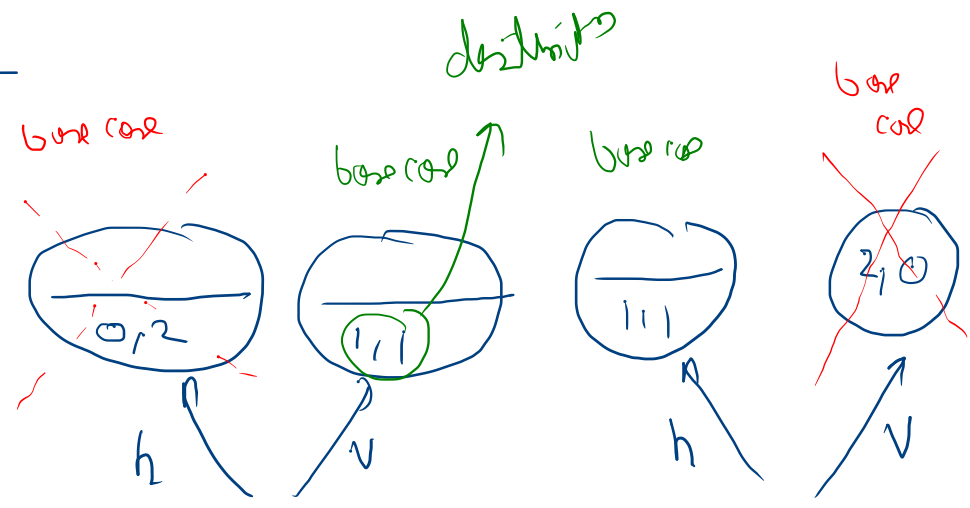
h v
v h

Small → Big

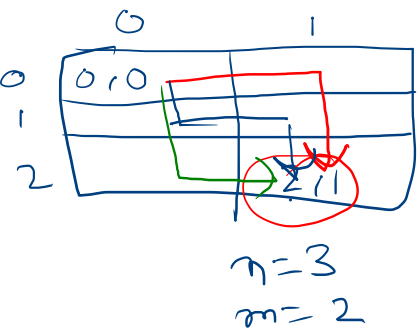
src → dest
dest → src

Big → Small

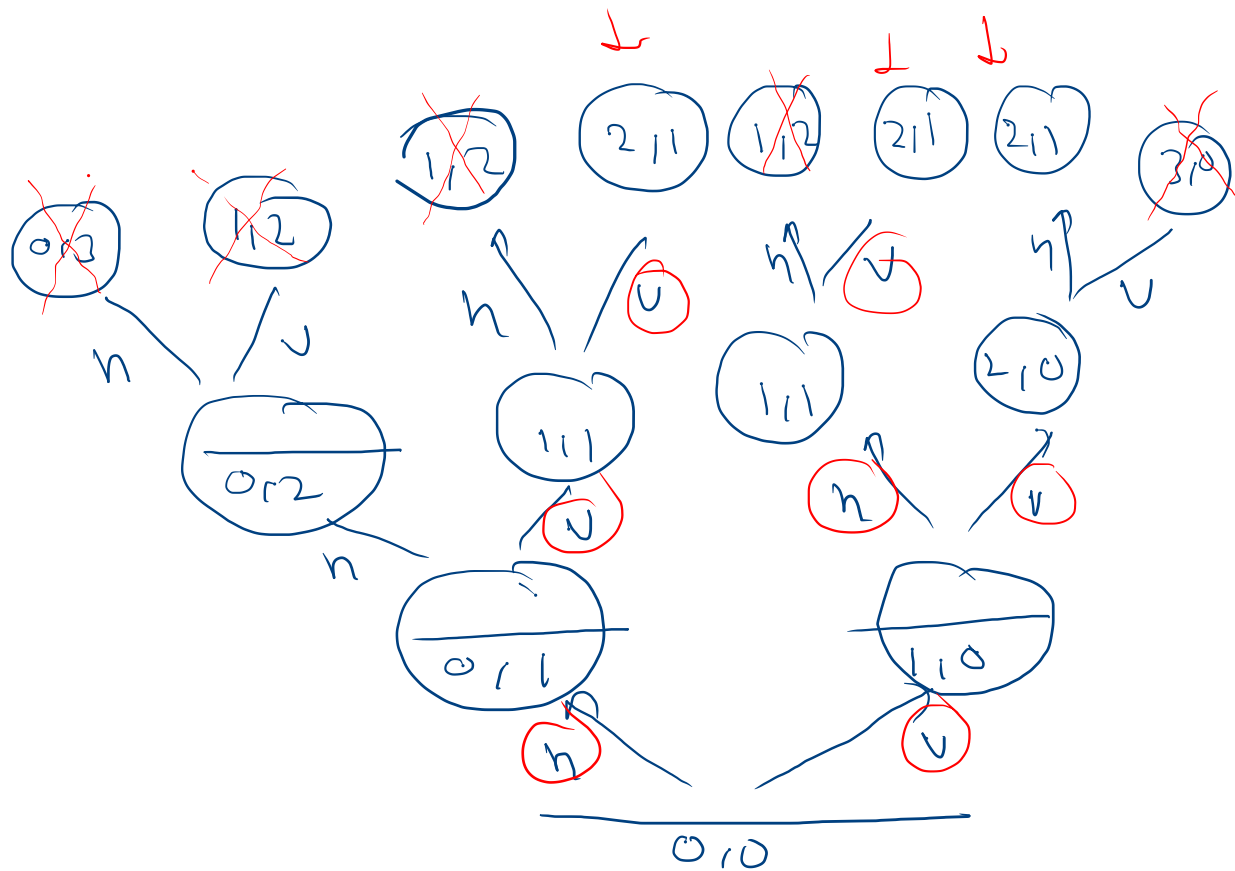
S_i, S_j



on, h, v, v



hvv
 $v hv$
 vvh



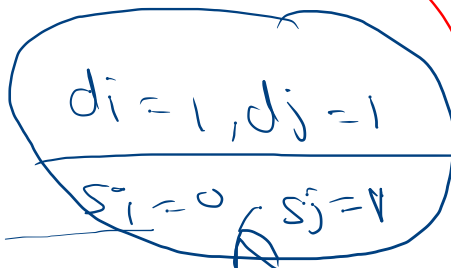


faith
method

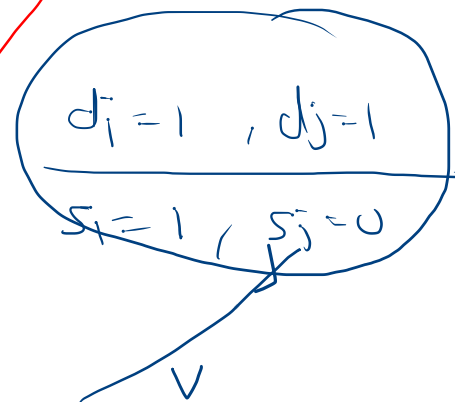
maze = 2, 2

src, dest

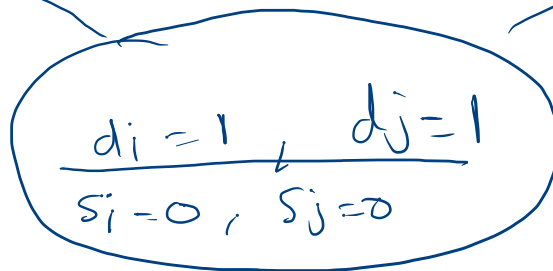
[V]



[h]



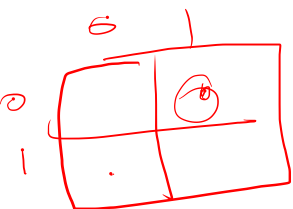
h



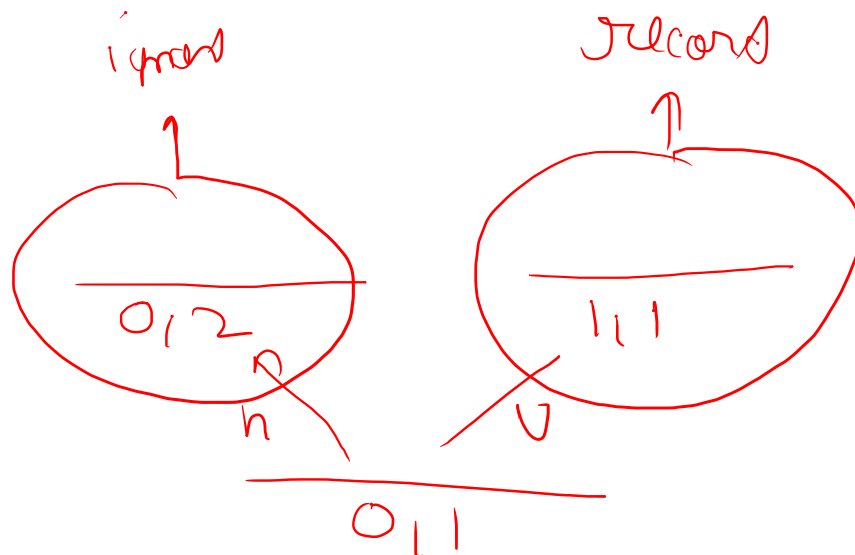
o

[hv]

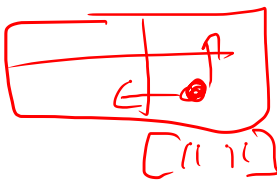
[vh]



base
case



base case \rightarrow ignore $\rightarrow s_i \geq n, s_j \geq m$

base case \rightarrow record $\rightarrow s_i == d_1 \text{ \& \& } s_j == d_j$ 
[1, 1]