

Most Important Question on Recursion.

⇒ Popularly asked in Interview.

Q.7

Point all the subsequences -

str = $\bar{a}\bar{b}c$

$f(abc)$

\hookrightarrow all the subsequences -

abc bc c

ab

b

ac

a



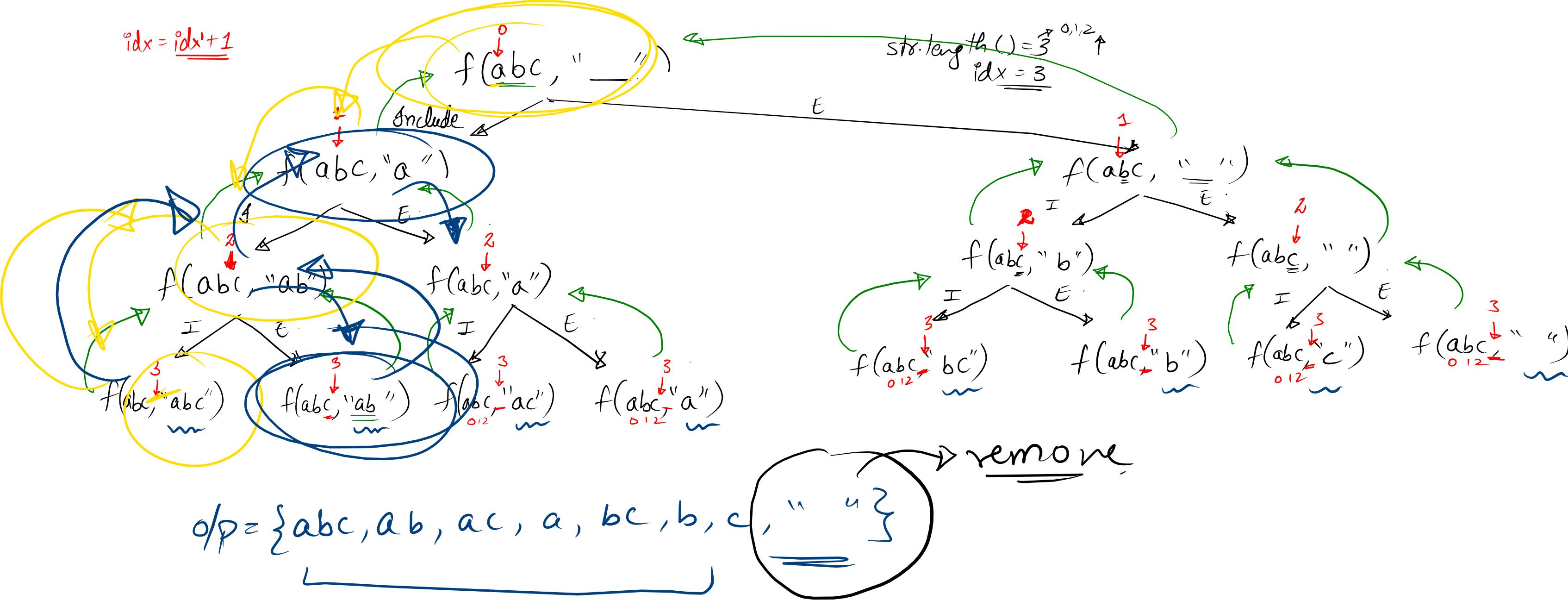
$f(\underbrace{abc}, \underbrace{''}_{\text{Input}} \quad '' \quad)$

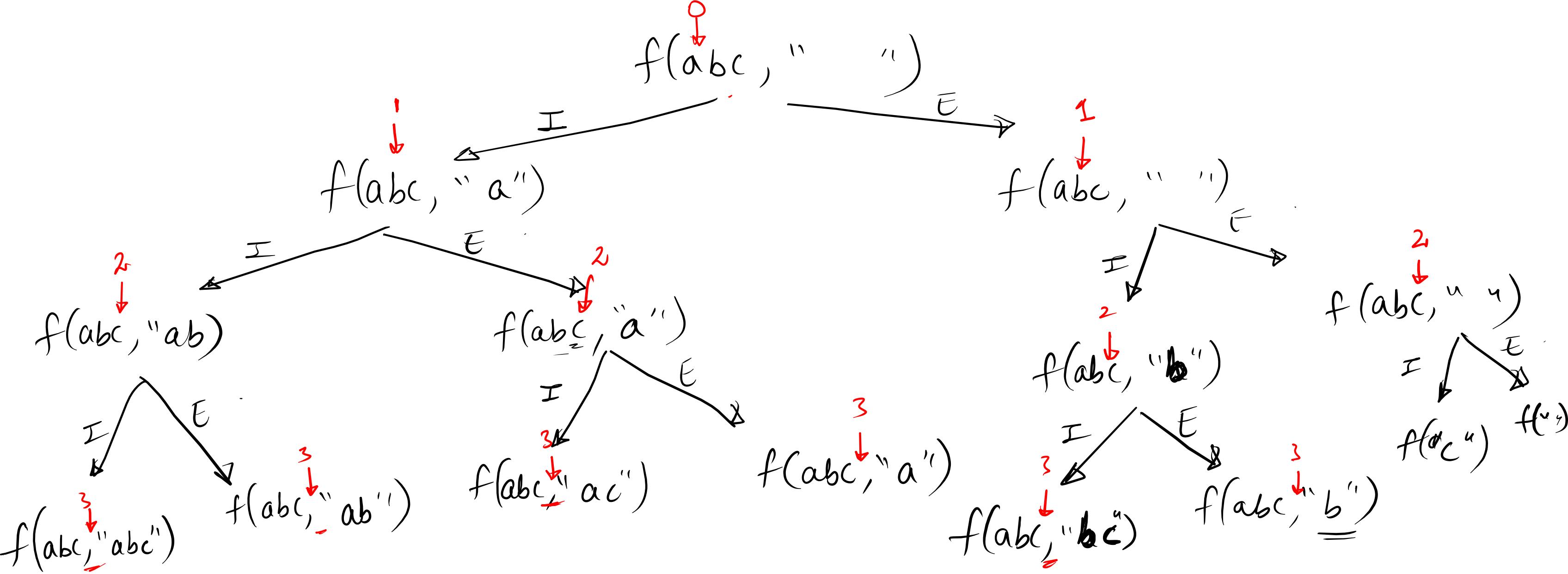
Input

Output / result -

from the given str -

$$\text{idx} = \underline{\text{idx}} + 1$$





```
static void solve(string str , String output , int idx

    //base case
    if(idx == str.length()){
        if(!output.isEmpty()) { // non Empty string p
            System.out.print(output+" ");
        }
        return ;
    }

    //process
    //step 1 include function call
    char ch = str.charAt(idx);
    solve(str,output + ch , idx +1 ) ;

    //step 2   exclude function call
    solve(str,output , idx+1) ;
```

Exclude

③ $f(s, \underline{\text{output}}, \text{idx}+1)$

Include

② $f(s, \text{output} + ch, \text{idx}+1)$

Base

① if ($\text{idx} == \text{str.length}()$) {
 ③
 s.o.p(output);
 return;
}

Q.) No. of ways to form Natural No.

$$f(N, 1)$$

$6 \Rightarrow (1, 2, 3)$

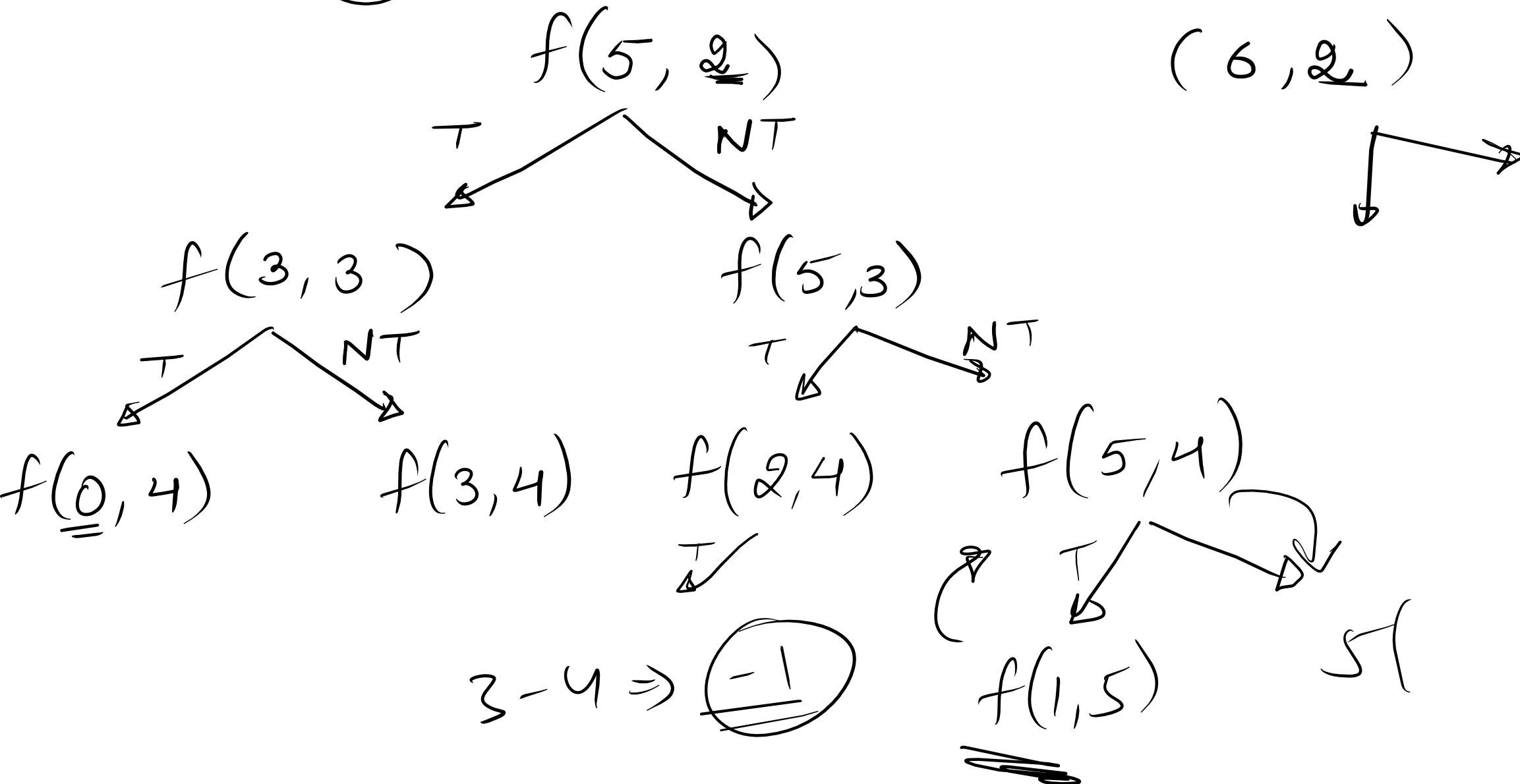
same meaning
 $\text{cnt}++$

$6-1-2-3 \Rightarrow \underline{\underline{0}}$

$f(6, 1)$
Take \swarrow \searrow Not Take

Cnt = 0

$f(j, i)$



```

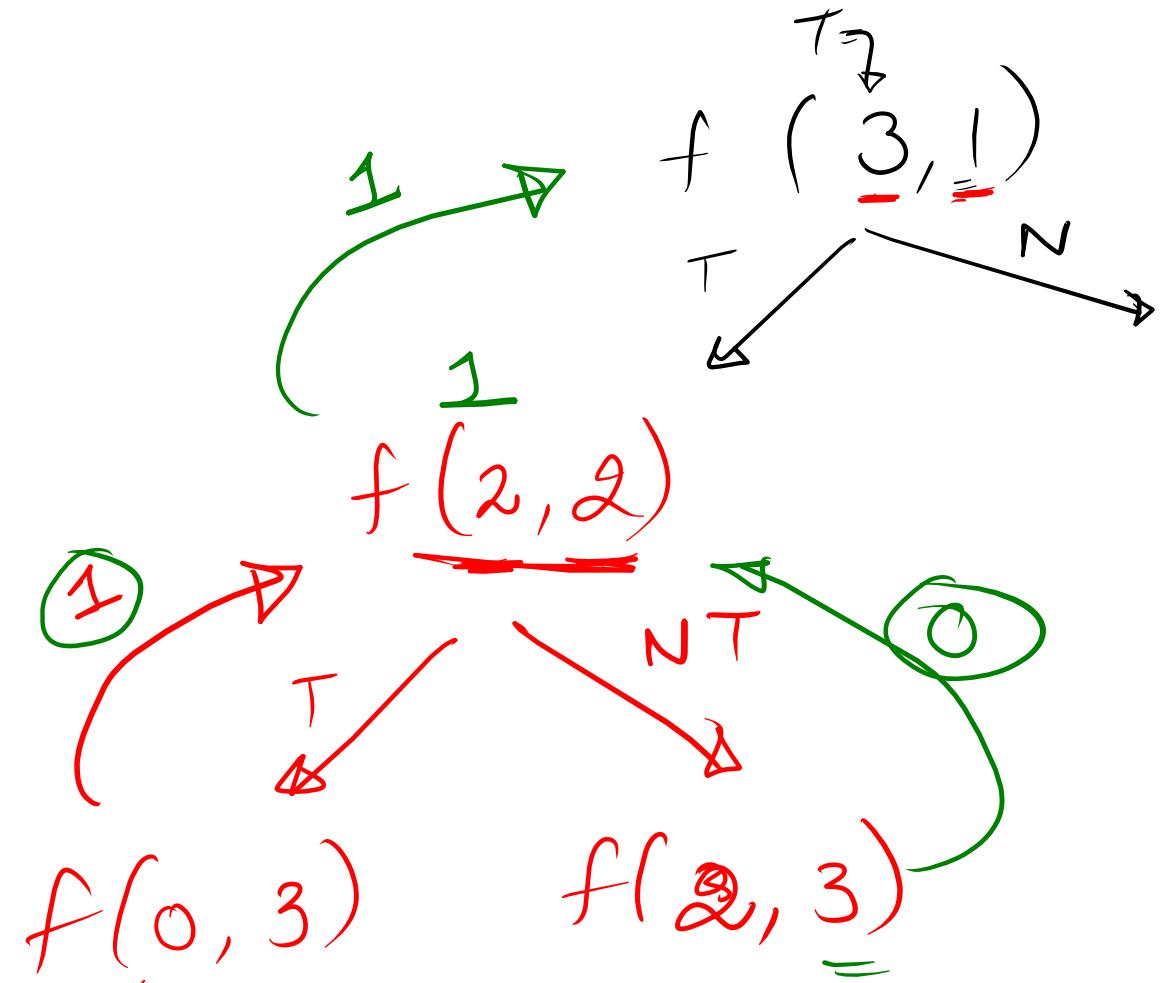
public static long ways(int Target, int i){
    //base case
    if(Target == 0) {
        return 1 ;
    }
    if(Target < i )  return 0 ;

    //process
    long take = ways(Target-i,i+1) ;    // we a

    long NOTtake = ways(Target,i+1);

    //return
    return take + NOTtake ; // not subtractin
}

```

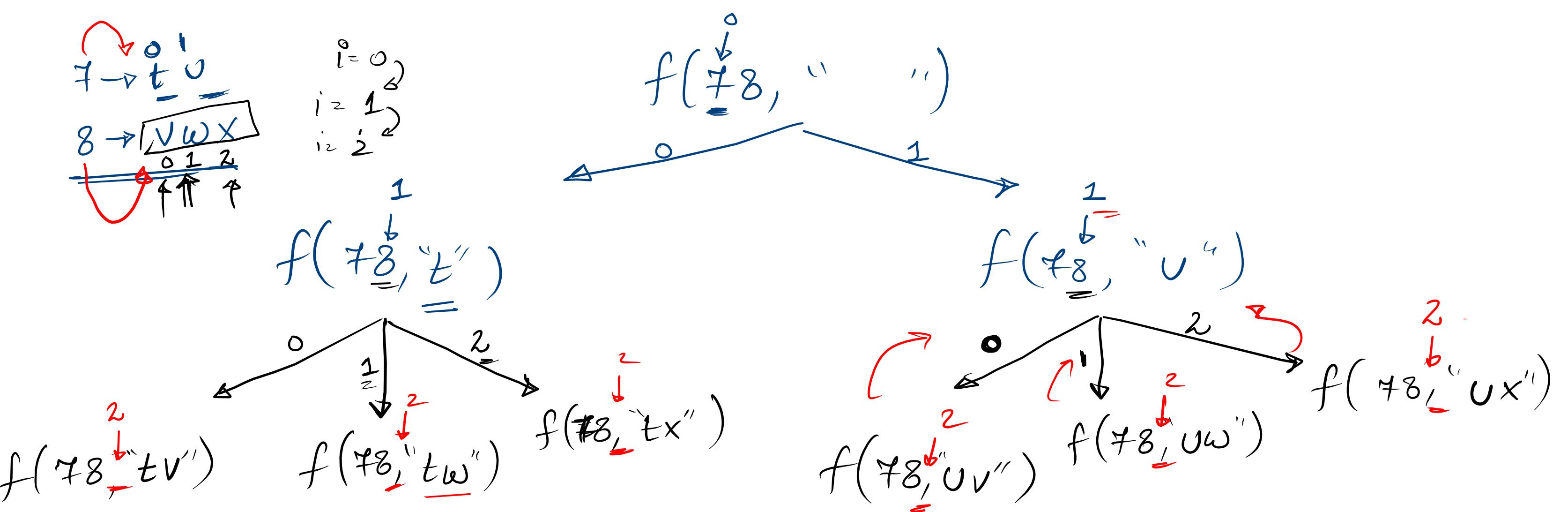


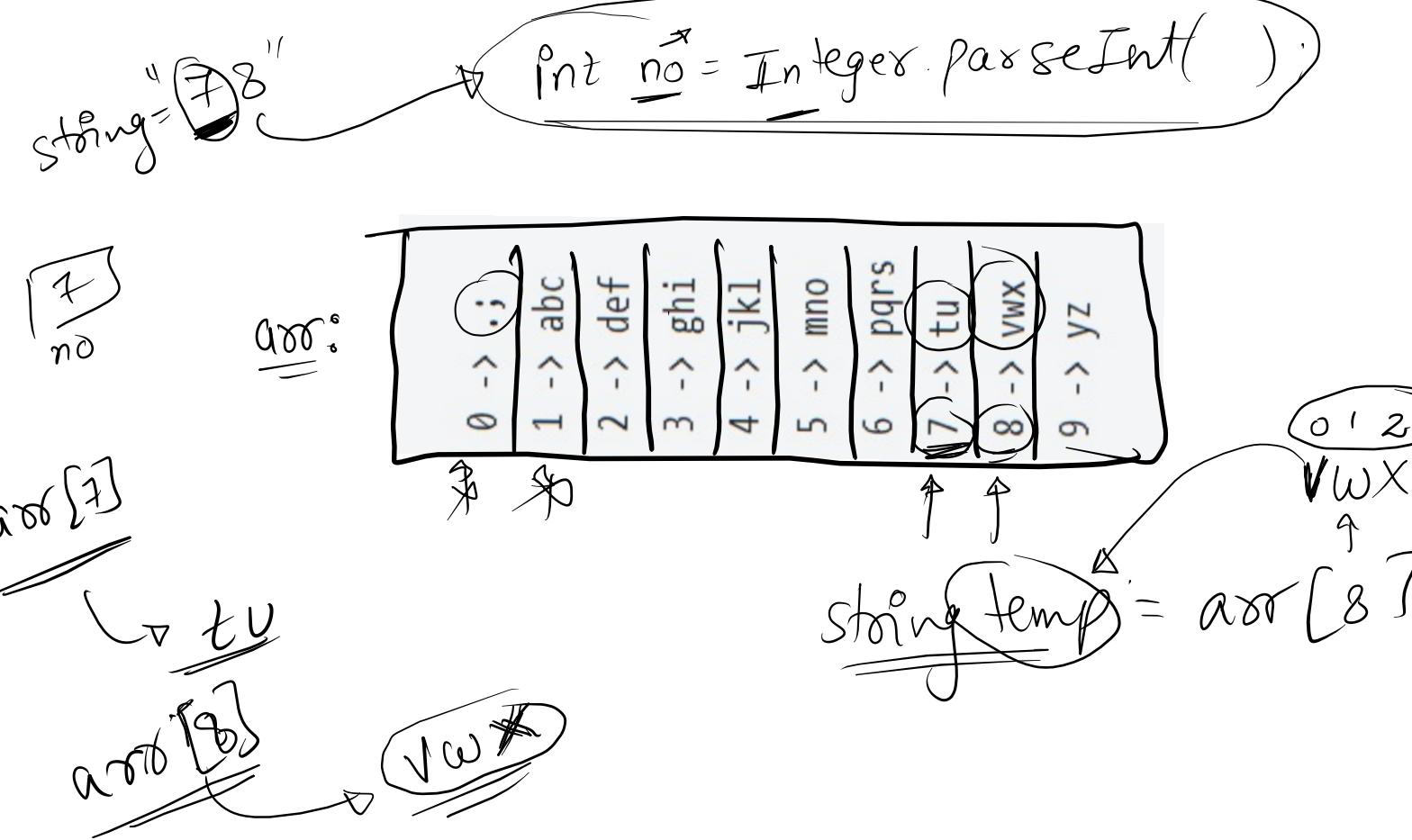
Q. Keypad combination

i/p \Rightarrow "78"

o/p \Rightarrow t^v, t^w, t^x , u^v, u^w, u^x

"7" \rightarrow "tu"
"8" \rightarrow "vwx"





~~for (int i = 0; i < temp.size()) {~~

char ch = temp.charAt(i);

f(ip, output + ch, i + 1);

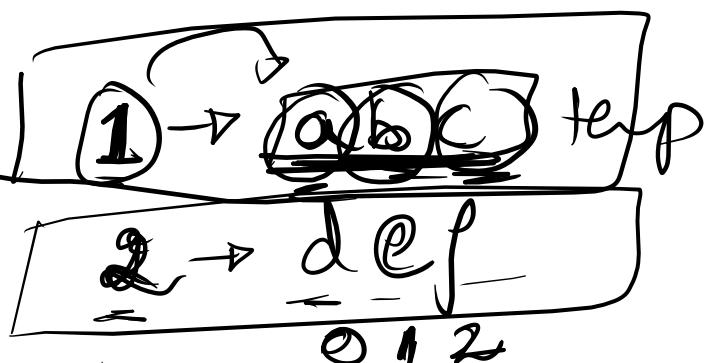
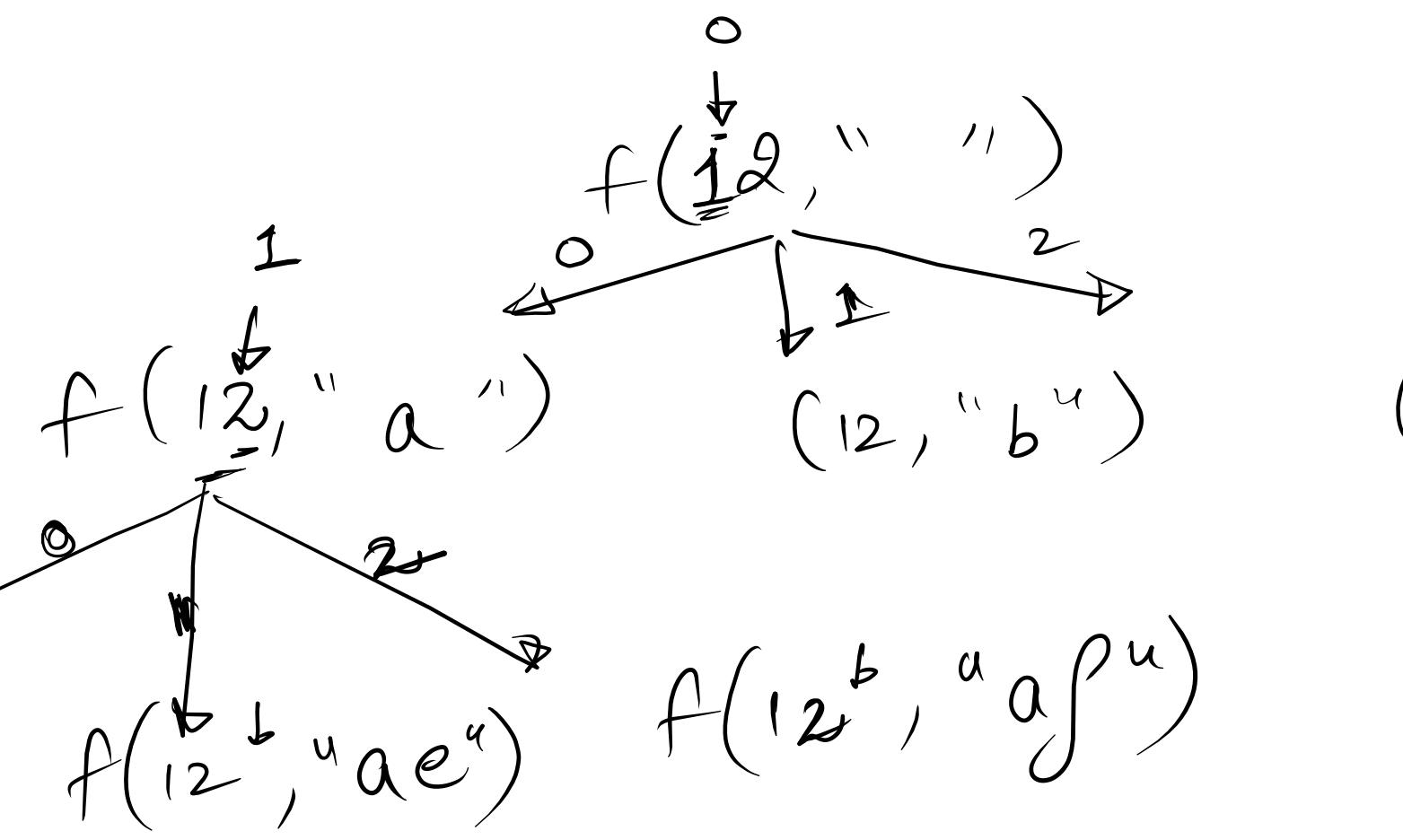
}

temp = 0 1 2
v w x

u v
fix new

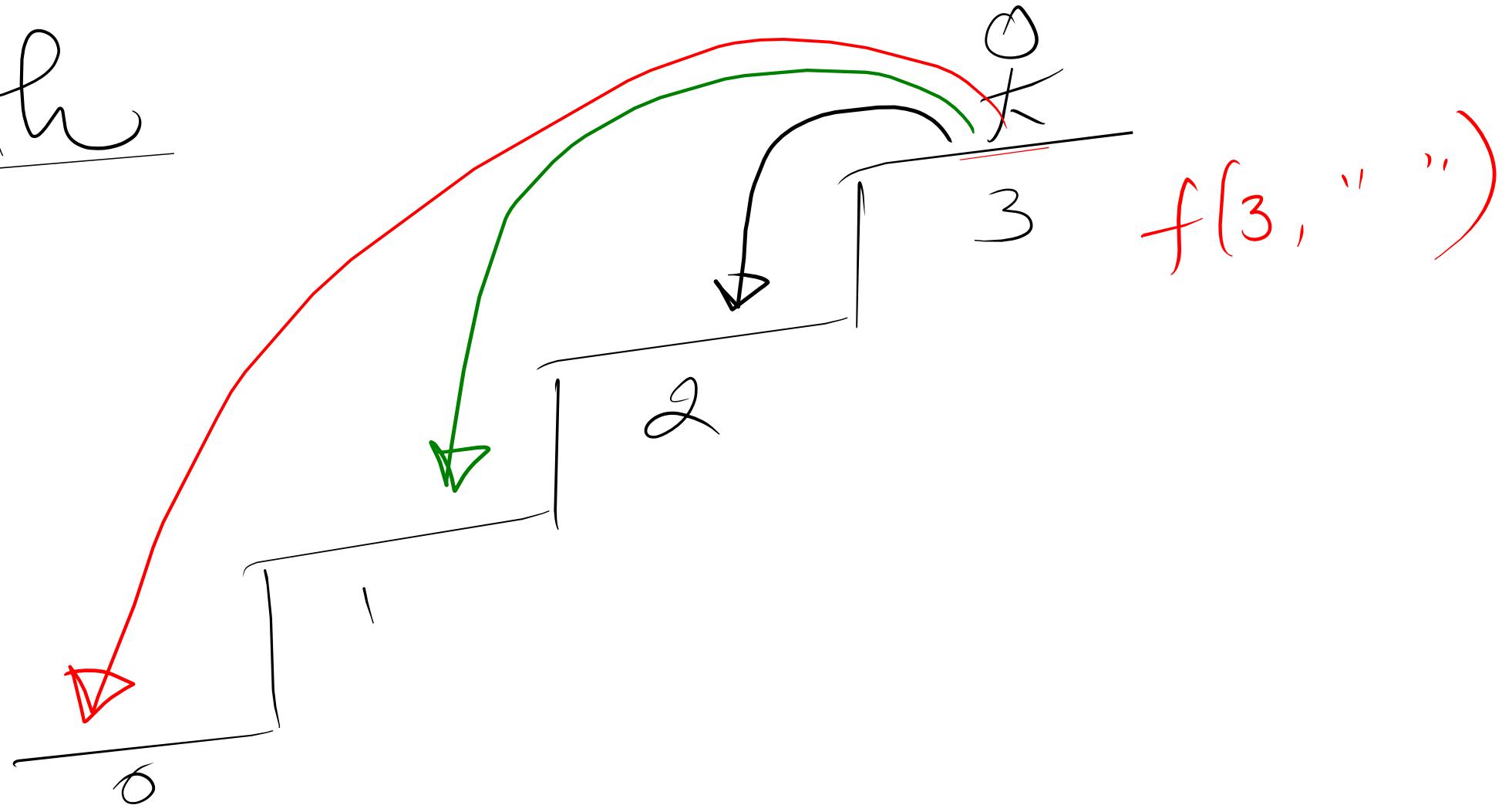
```
static void solve(String str , String output,int idx , String[] arr){  
//base case  
if(idx == str.length()){      // (out of str length idx -->)2 == 2(length of str)  
    System.out.println(output);  
    return ;  
}  
  
//process  
int number = str.charAt(idx)-'0' ; // '7' - '0' ---> 55 - 48 ==> 7(int)  
String temp = arr[number]; // temp = "tu" ->> length = 2      for loop i = 0 , 1  
  
for(int i = 0 ; i<temp.length() ; i++){  
    char ch = temp.charAt(i);  
    solve(str,output+ch,idx+1,arr);  
}  
}
```

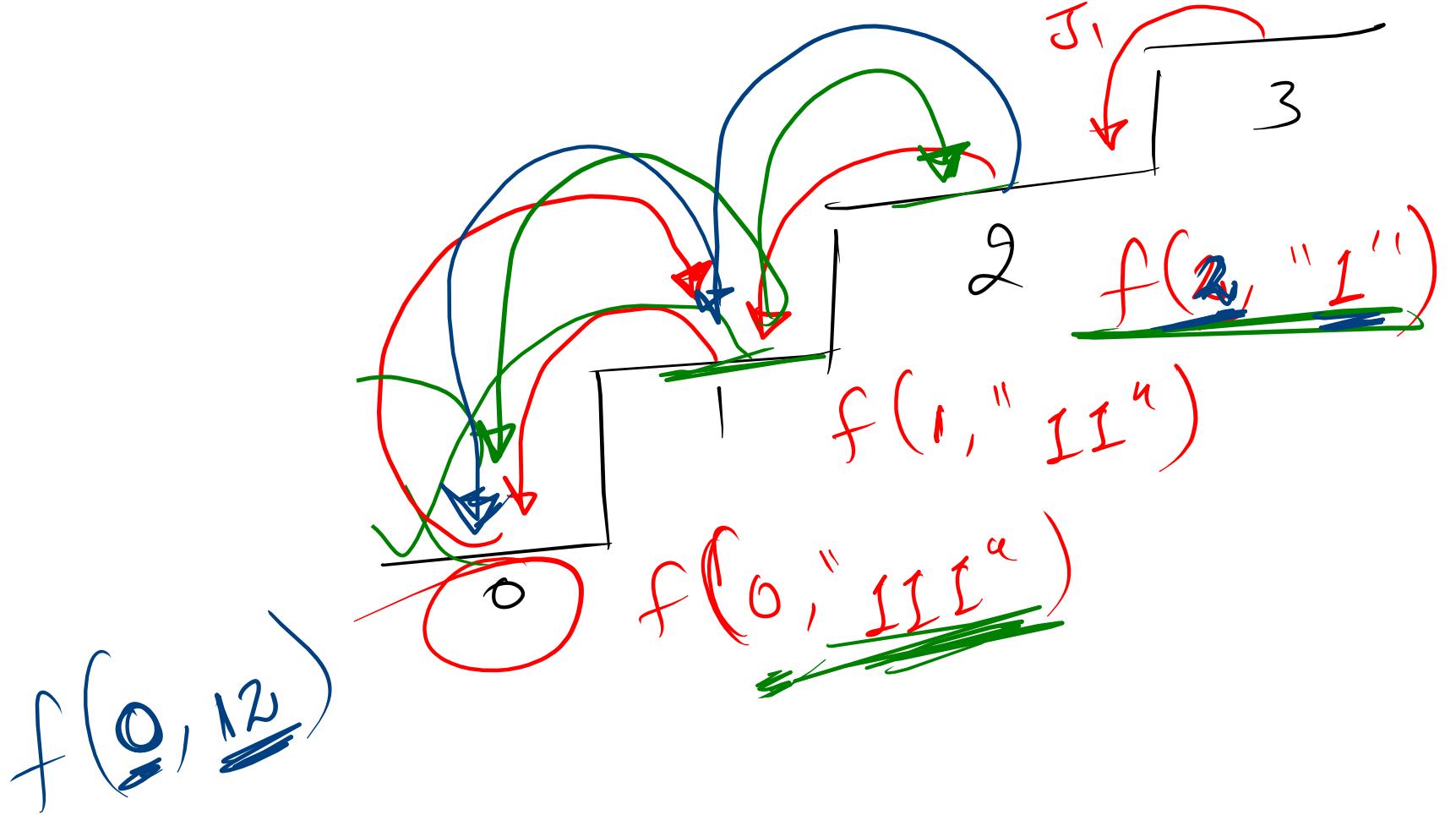
$\text{if } p = "12"$



Q) Point stairPath

$n = 3$





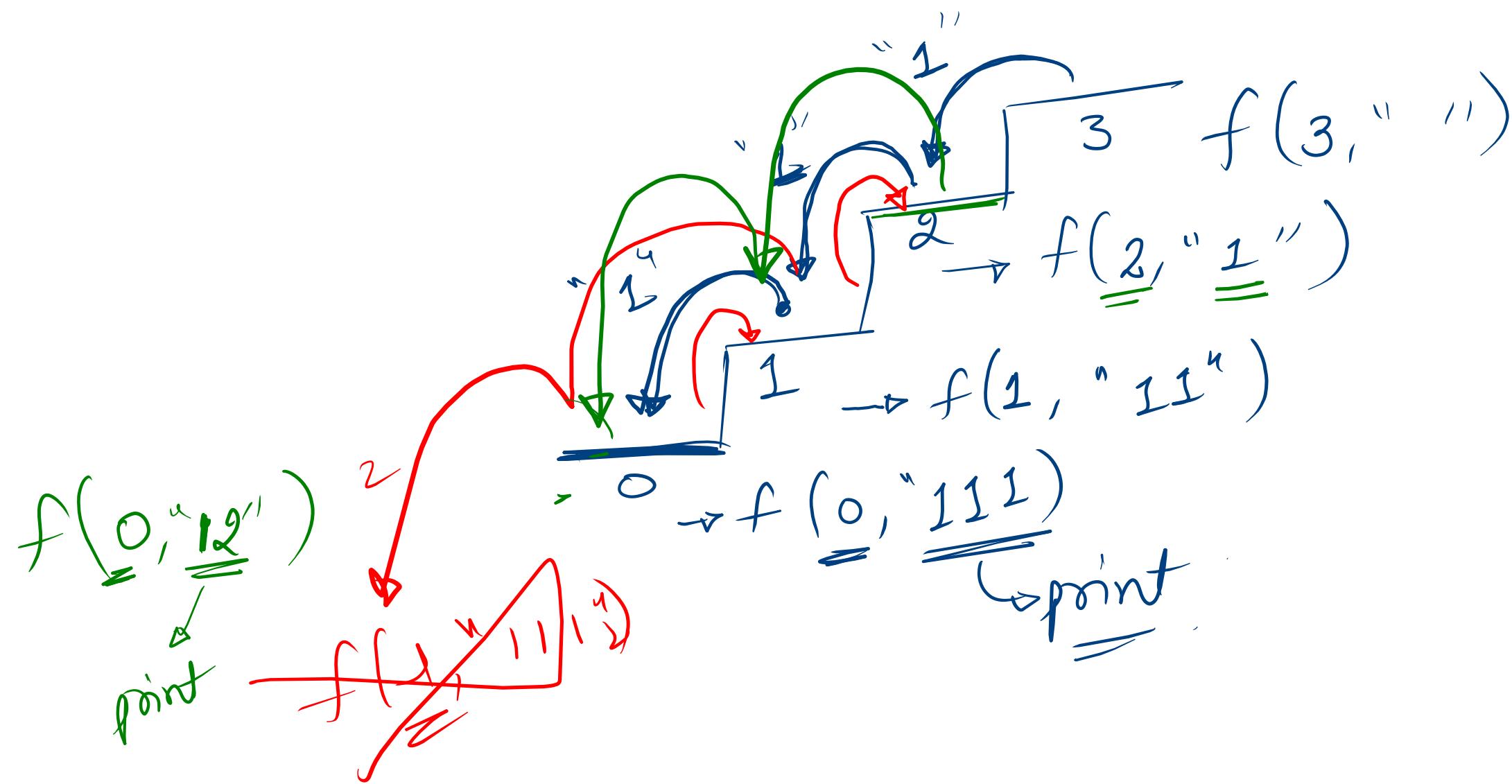
if($n < 0$) return;
 if($n == 0$) S.O.P(O.P)

$f(\underline{3}, " = ")$

$3 - 1 \Rightarrow 2$

✓ Jump 1 = op + "1"
 ✓ Jump 2 = op + "2"

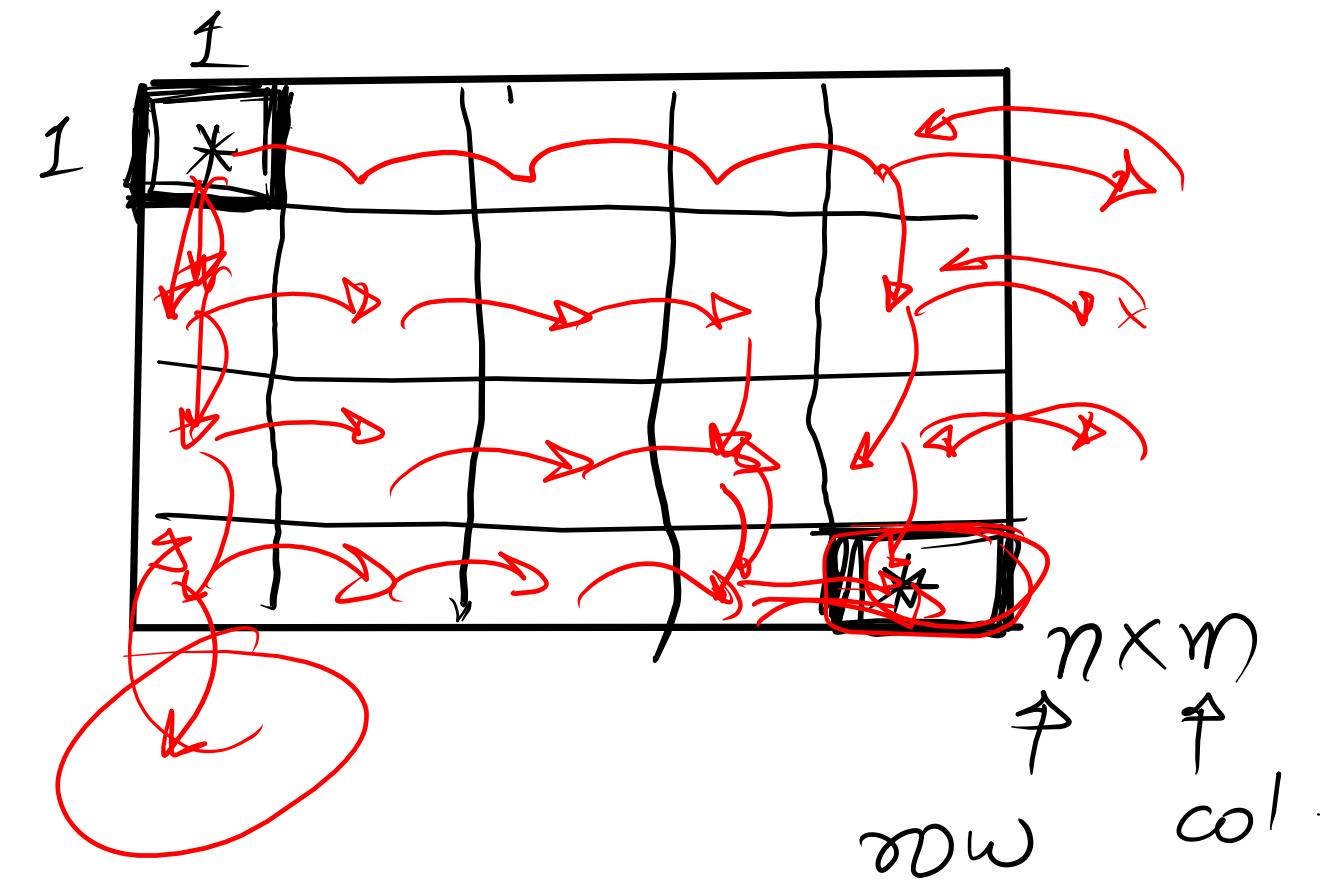
Jump 3, op + "3",



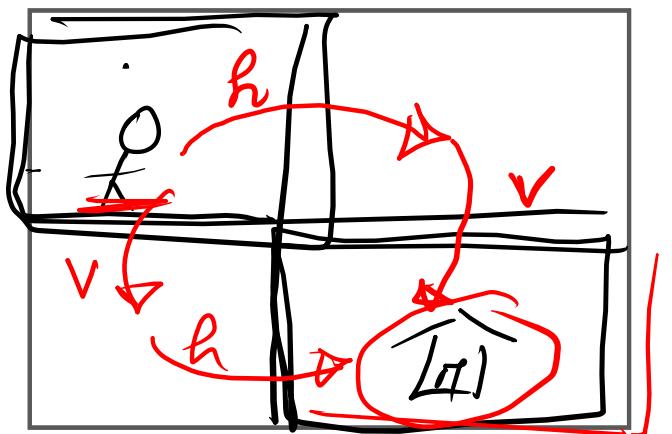
String Palindrome

```
static boolean solve(String str,int left , int right){  
    if(left  >= right) return true ;  
    if(str.charAt(left) != str.charAt(right)) return false ;  
  
    return solve(str,left+1,right-1);  
}  
  
static boolean isPalindrome(String str){  
    int left = 0 ;  
    int right = str.length()-1;  
    return solve(str,left , right);  
}
```

a Maze Problem



$[2, 2]$

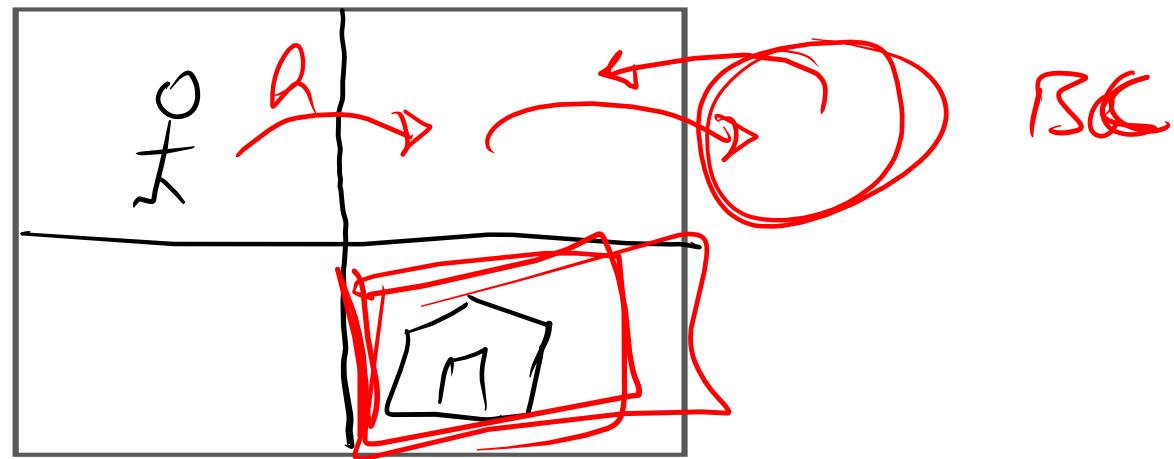


"hv" ✓
"vh" ✓
=====

You can take only 2-moves

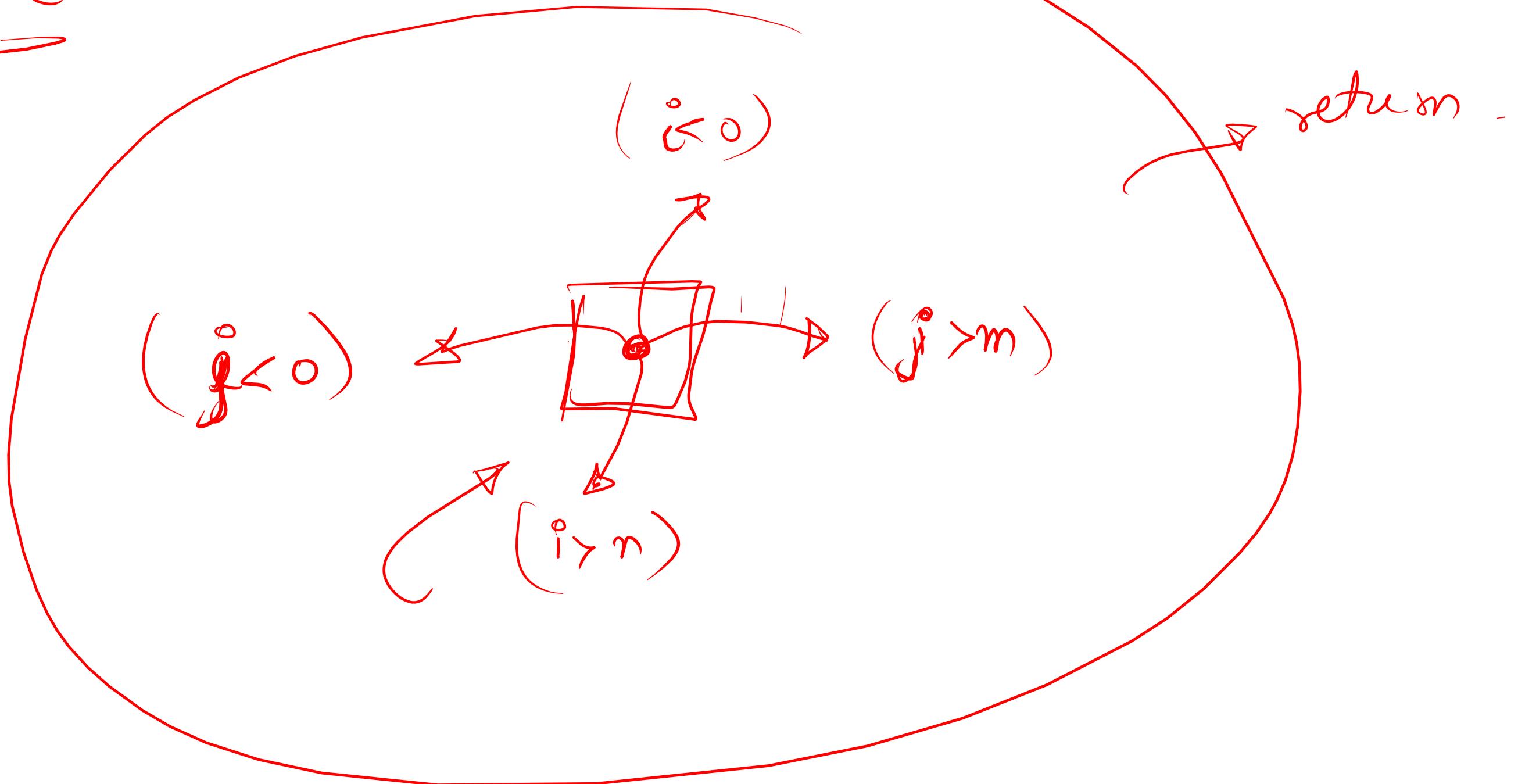
→ horizontal → 'h'
→ Vertical → 'v'

if()



if ($i = n \& \& j == m$)
 Point;
 setem;
}

Base Case



① Base Case:

```
[ if( T == 0)      cnt++;  
  if( T < 0)      return;  
                  =
```

② process

③ ↙

Take = solve ($T - i, i+1$);

② ↙

Nottake = solve ($T, i+1$);

return Take + Nottake

TOH

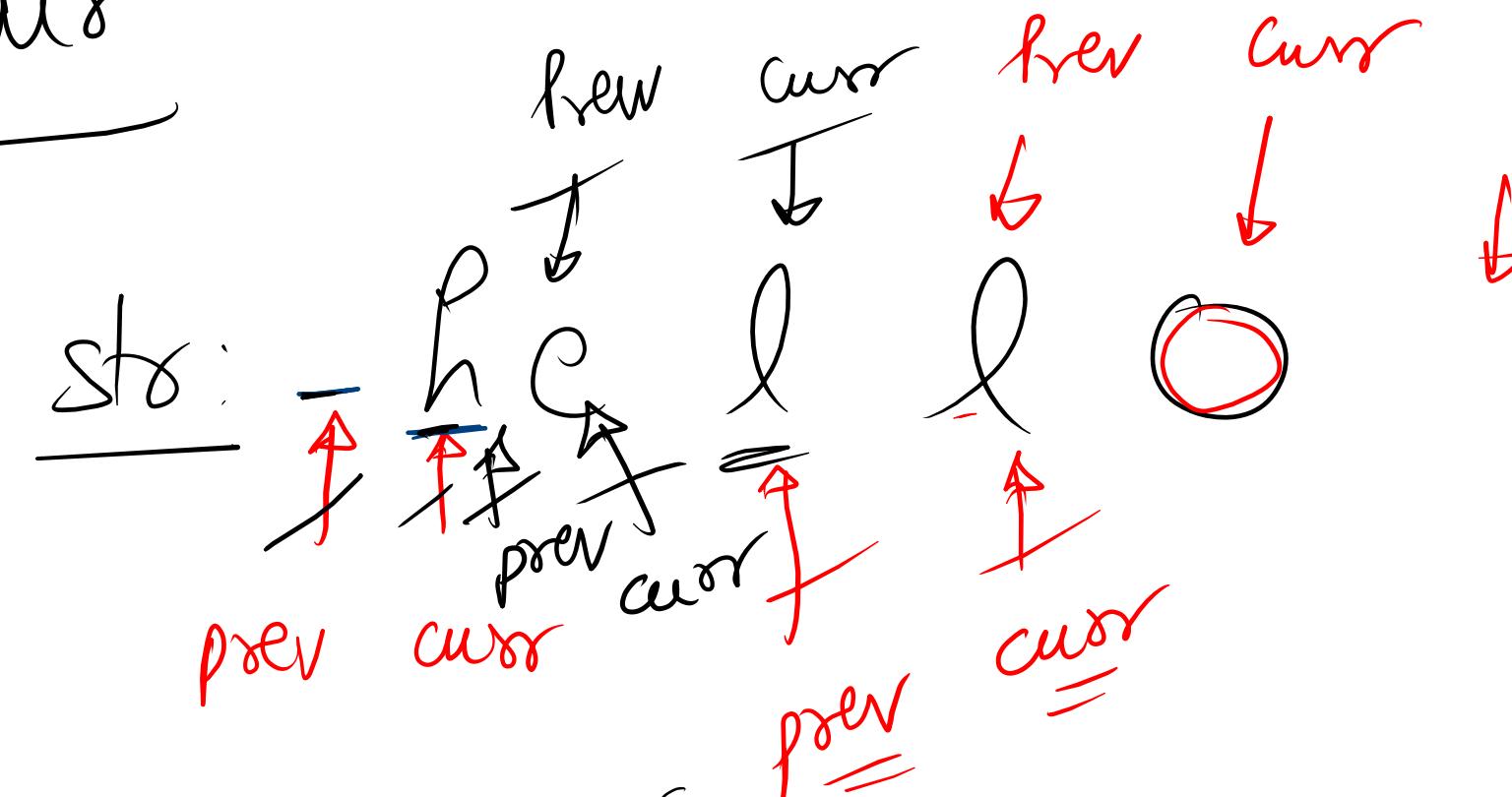
[https://course.acciojob.com/idle?
question=cb2cd5ce-46df-4bea-9ea6-12
902858badf](https://course.acciojob.com/idle?question=cb2cd5ce-46df-4bea-9ea6-12902858badf)

~~* a Maze Problem~~

SHU

Doubt?

* Pair Stair



if(prev == curr) {

opt '*' + curr.

}

hello → hel*lo
i/p: OPT

dp = " hel*lo "

else {

opt + curr

}

```
static String solve(String str , String output , int idx,char prev){

    //base case
    if(idx == str.length()){
        return output ;
    }

    // process
    char curr = str.charAt(idx);
    if(curr == prev){
        return solve(str, output+'*'+curr,idx+1,curr);
    }
    else{
        return solve(str,output+curr,idx+1,curr);
    }

}

static String PairStar(String str) {
    char prev = '_';
    return solve(str,"",0,prev) ; // solve(string str , output , idx = 0 )
}
```

Q. Maze Problem

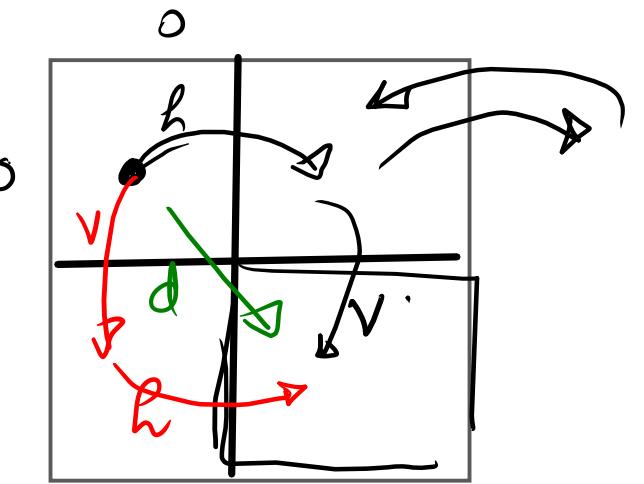
Input

$$n = 2$$

$$m = 2$$

Output

$h1v1$
 $v1h1$
 $d1$

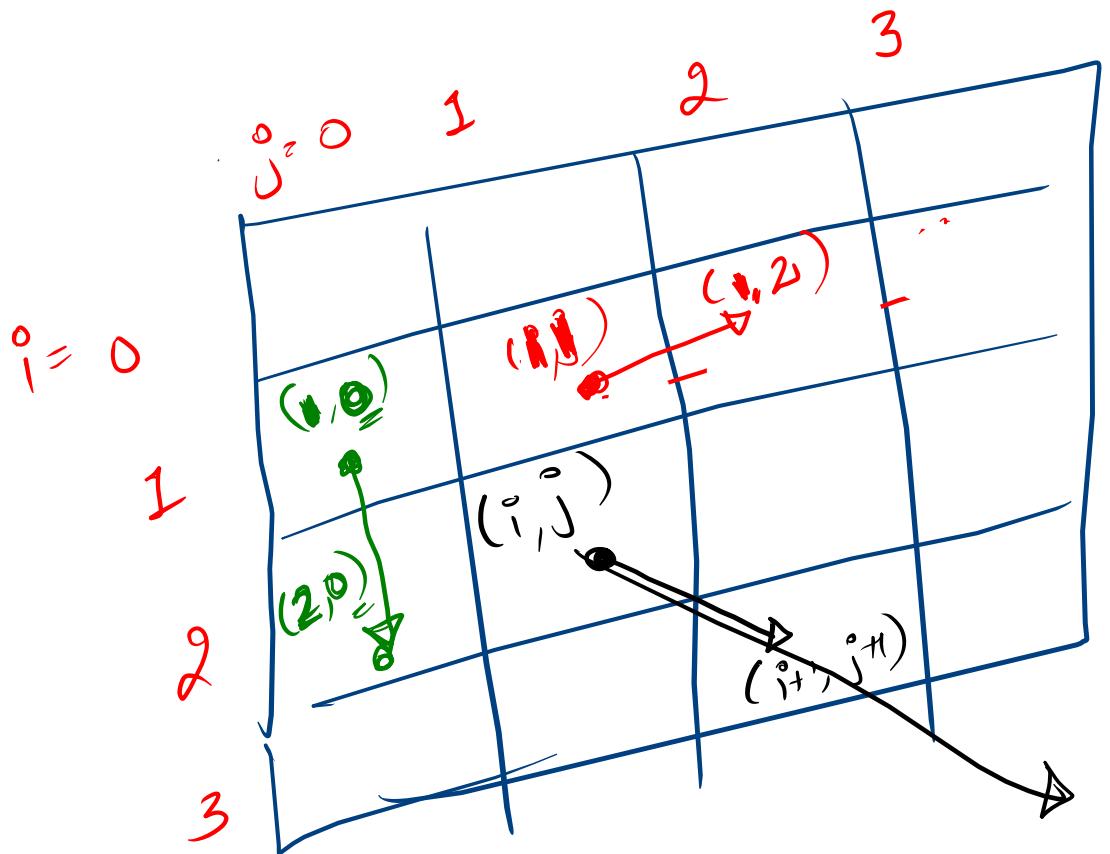


$h1v1$

$v1h1$

$d1$

Index game



for Horizontal move

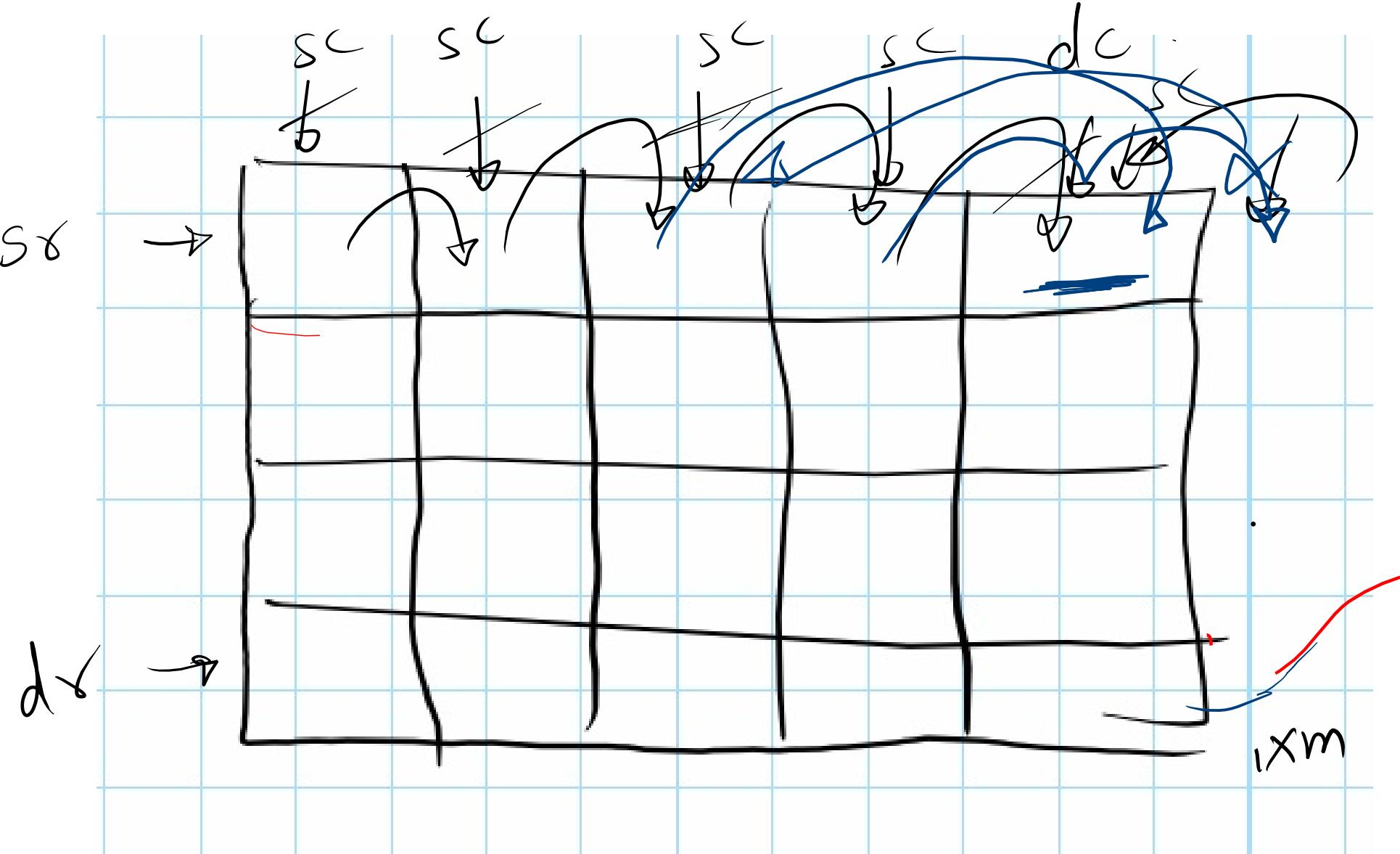
$$(i, j) \rightarrow (i, j+x)$$

for Vertical move

$$(i, j) \rightarrow (i+x, j)$$

for diagonal move

$$(i, j) \rightarrow (i+x, j+x)$$



for

$h \Rightarrow 1 \dots d$

$sc + 1$

$sc + 2$

$sc + 3$

$sc + 4$

$sc + 5$

$sc, sc + h$

$f_i x$

update

if ($sc == n \& sc == m$) {
 point(answer);
 return;
}

Base case

~~#~~ Towers of Hanoi