Most Important Question on Recursion.

$\Longrightarrow$ Popularly asked in Interview.

Q> Print all the subsequences -

str = abc

abc bc c
ab b
ac
a

f(abc)

↳ all the subsequences - from the given str.

f(abc, " ")

input output/result -

idx = idx'+1

f(abc, "")

include

E

str.length() = 3

idx = 3

f(abc, "a")

f(abc, "ab")

f(abc, "a")

f(abc, "abc")

f(abc, "ab")

f(abc, "ac")

f(abc, "a")

f(abc, "b")

f(abc, "c")

f(abc, "")

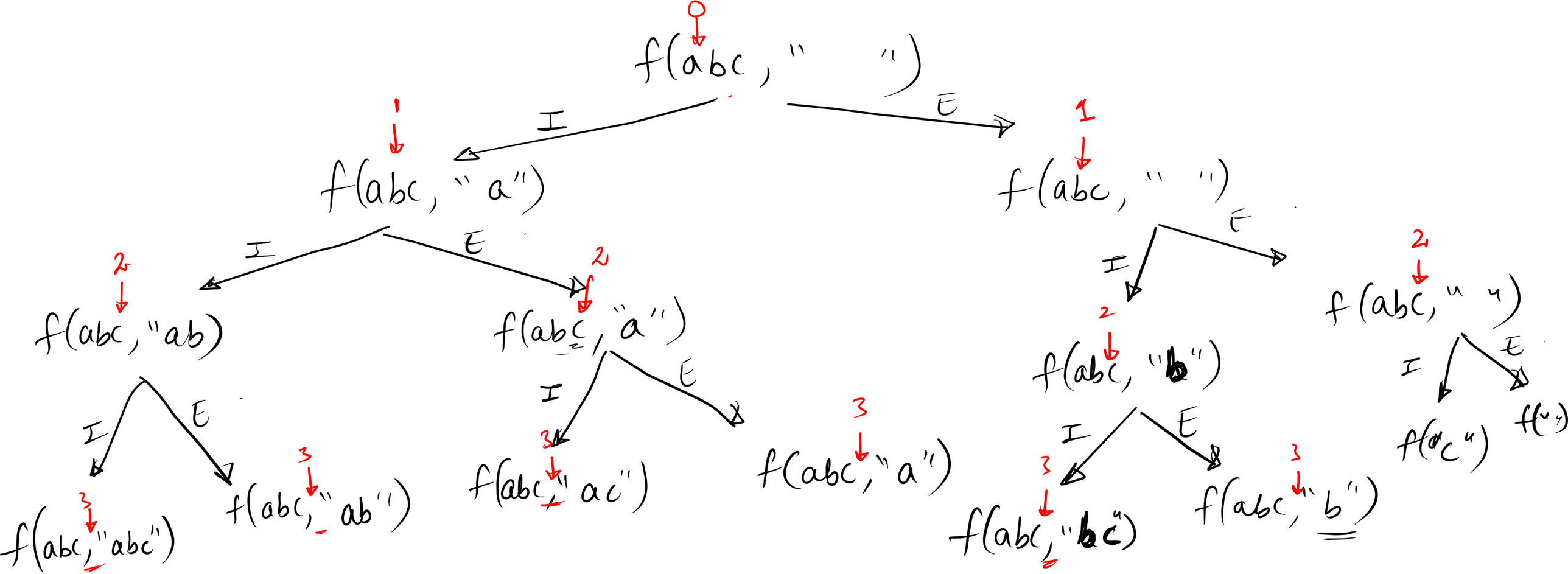
f(abc, "bc")

f(abc, "b")

f(abc, "")

remove

o/p = {abc, ab, ac, a, bc, b, c, ""}


f(abc, "")
I
E
f(abc, "a")
I
E
f(abc, "ab)
I
E
f(abc, "abc")
f(abc, "ab")
f(abc, "a")
I
E
f(abc, "ac")
f(abc, "a")
f(abc, "")
I
E
f(abc, "b")
I
E
f(abc, "bc")
f(abc, "b")
f(abc, "")
I
E
f(abc, "")
f(abc, "")
1
2
3

```java
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, output, idx+1)

Include

✓ ②

f(s, output + ch, idx+1)

Base

①

```
if(idx == str.length()) {
```

0,1,2 ③ ↔ ③

```
    S.O.P(output);
    return;
}
```

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

$f(N, 1)$

6 ⇒ (1,2,3)

6-1-2-3 ⇒ 0

same meaning cnt++

T

f(6, 1)

Take — Not Take

cnt = 0

f(J, i)

f(5, 2)

T — NT

f(3,3) — f(5,3)

(6,2)

T — NT

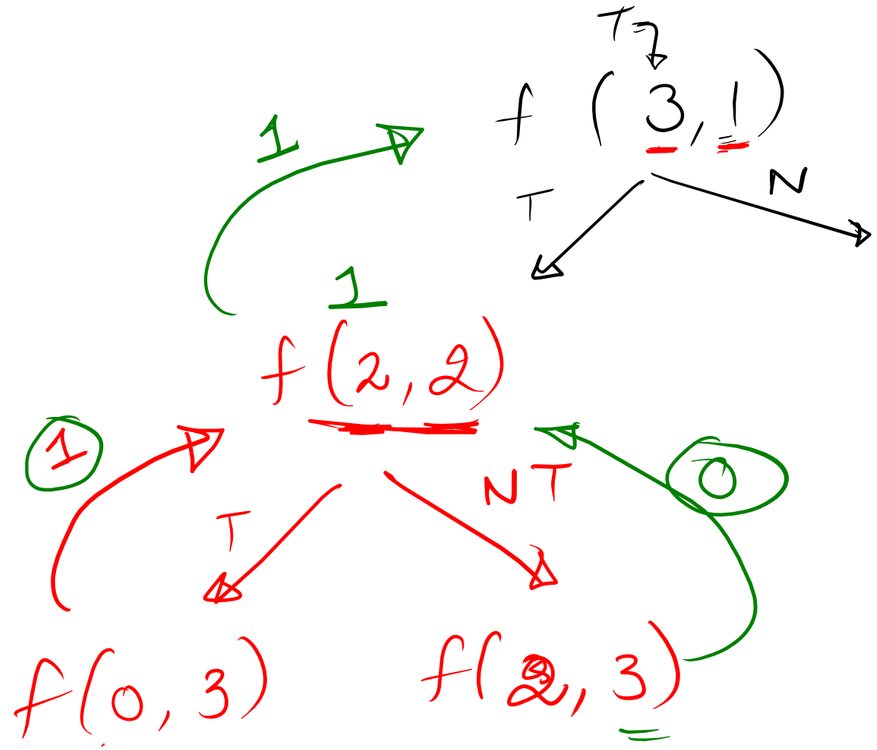
f(0,4) — f(3,4) — f(2,4) — f(5,4)

T — T

3-4 ⇒ -1

f(1,5)

```java
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);

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

}
```

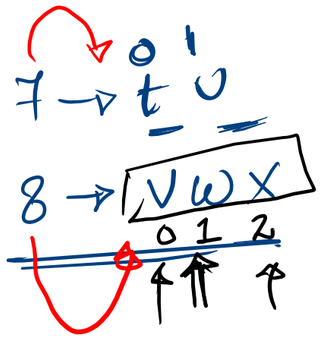


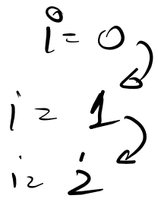
Q. Keypad combination

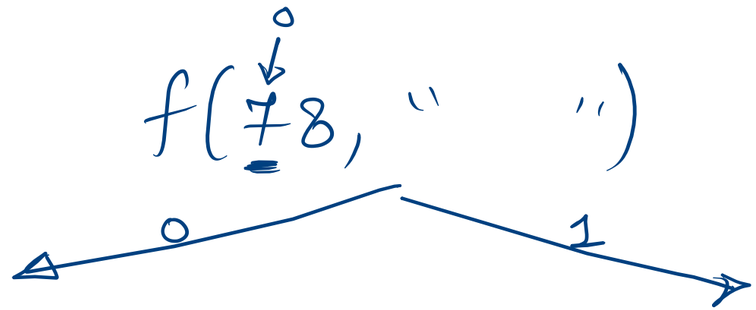
"7" → "tu"
"8" → "vwx"

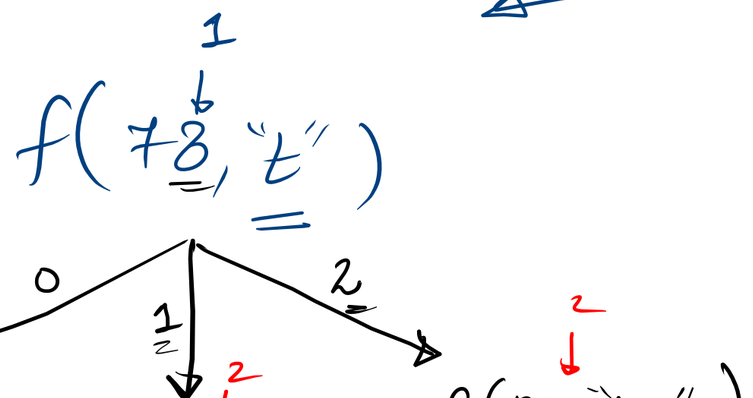
i/p ⇒ "78"

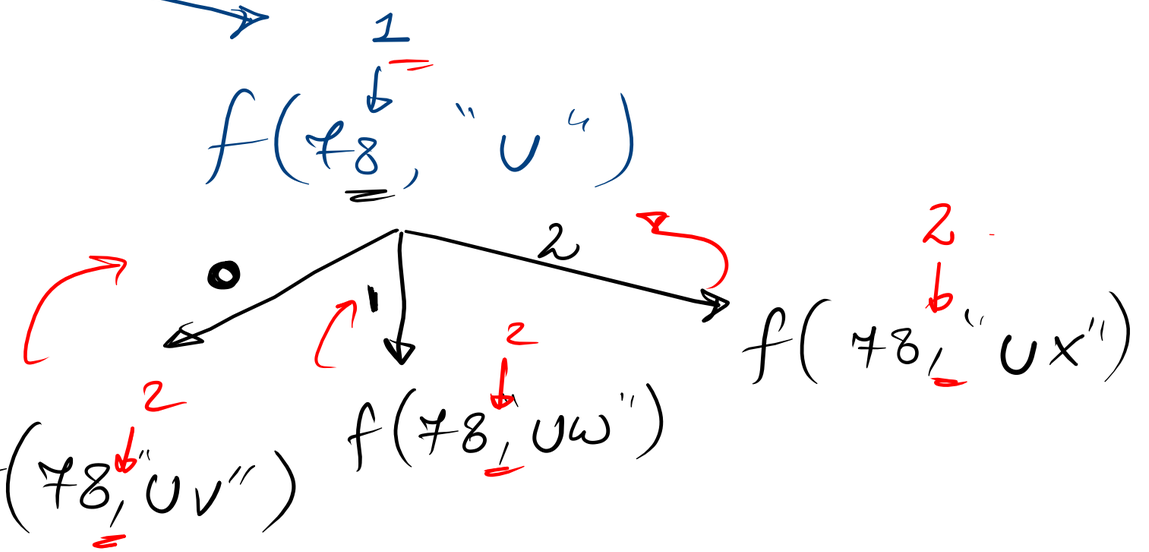
o/p ⇒ tv, tw, tx, uv, uw, ux


7 → t u
8 → v w x
0 1 2


i = 0
i = 1
i = 2


f(78, " ")
0
1


f(78, "t")
0
1
2
f(78, "tv")
f(78, "tw")
f(78, "tx")


f(78, "u")
0
1
2
f(78, "uv")
f(78, "uw")
f(78, "ux")

String = "78"

int no = Integer.parseInt( );

no = 7

arr:

0 -> ;	1 -> abc	2 -> def	3 -> ghi	4 -> jkl	5 -> mno	6 -> pqrs	7 -> tu	8 -> vwx	9 -> yz

arr[7]

↳ tu

arr[8]

vwx

0 1 2

vwx

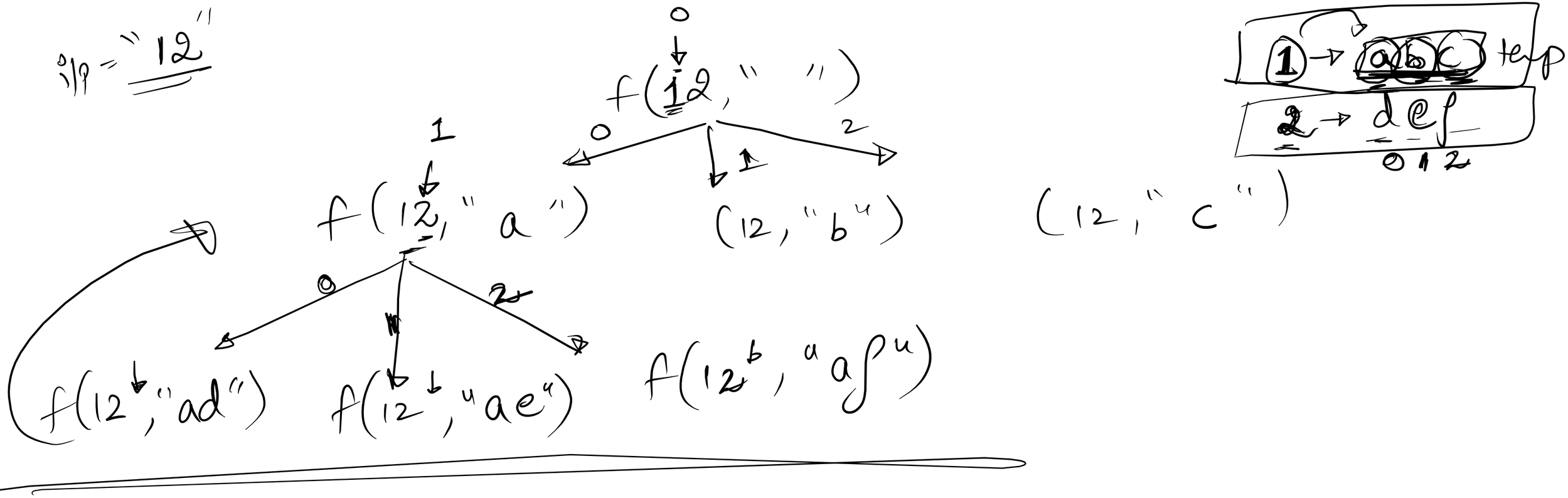
String temp = arr[8]

```
for(int i = 0; i < temp.size()) {
    char ch = temp.charAt(i);
    f( ip, output + ch, i+1);
}
```

temp = V W X
0 1 2

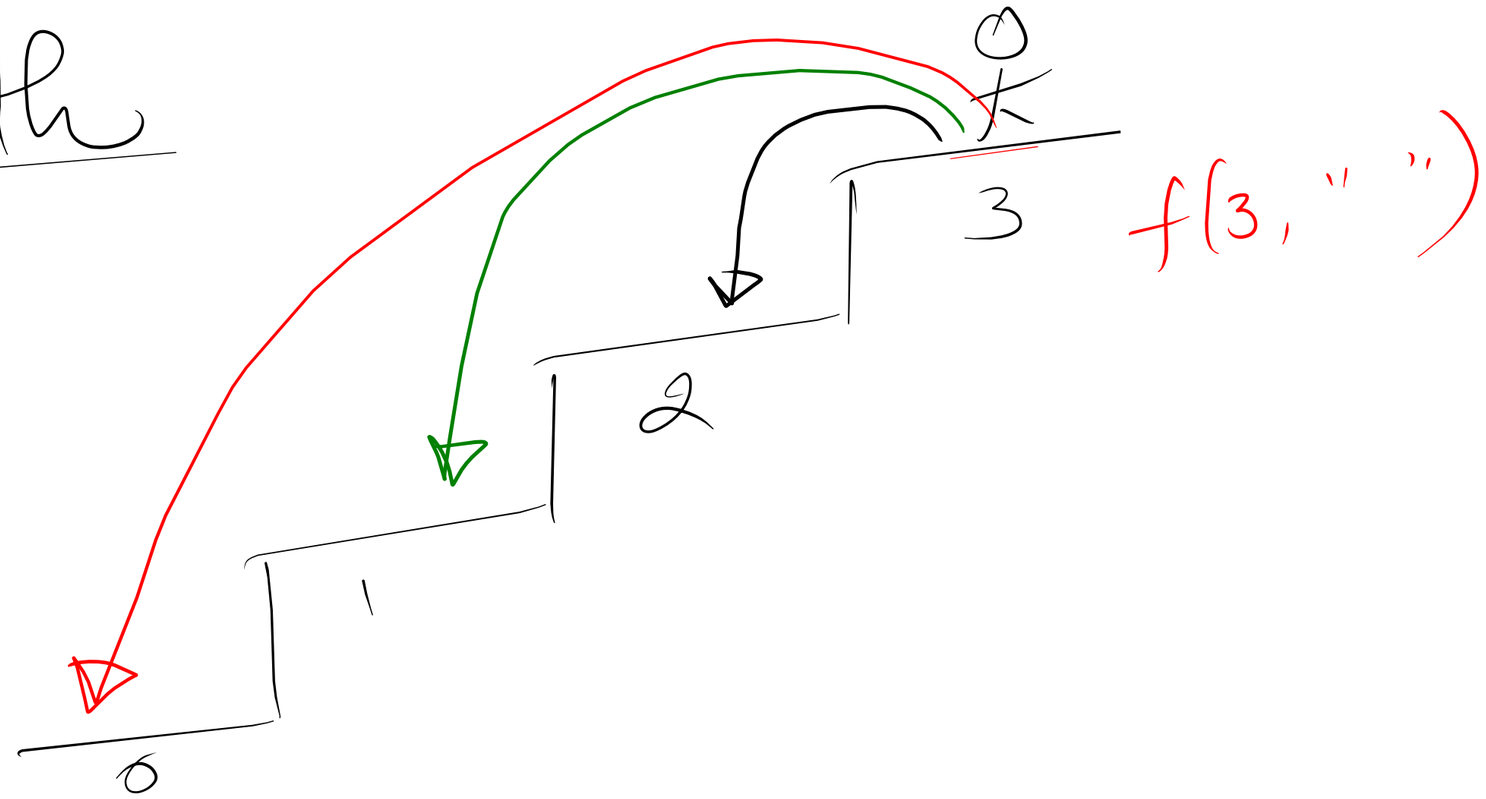
```java
static void solve(String str , String output,int idx , String[] arr){
//base case
if(idx == str.length()){     // (out of str length idx -->)2 == 2(length of str)
    System.out.println(output);
    return ;
}

//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);
 }

}
```

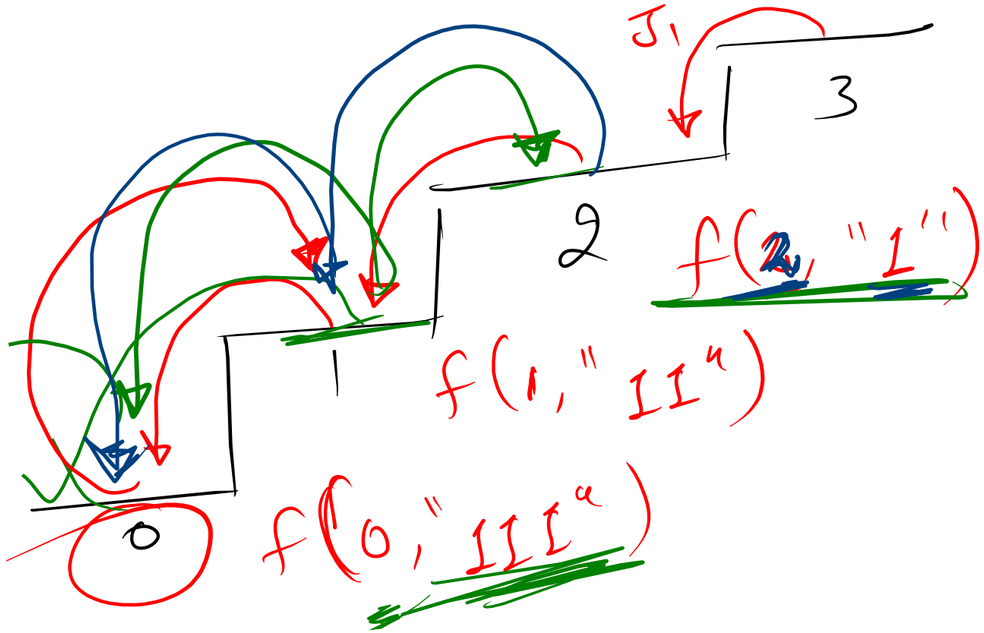
i/p = "12"



Q.) Print stairPath

n = 3



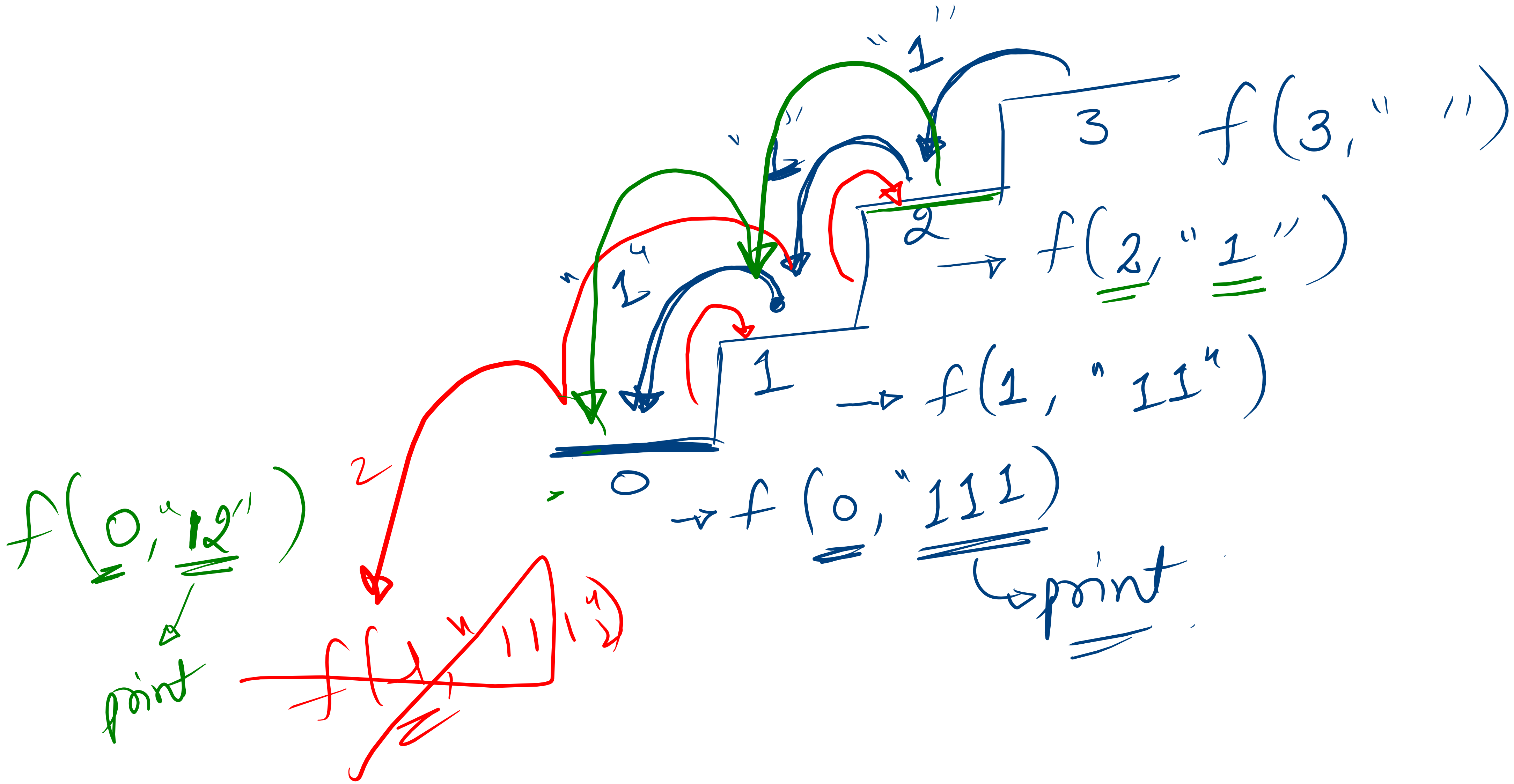



```
if(n<0) return;
if(n==0) stop(OIP)
```

Jump 1 = opt "1"

Jump 2 = opt "2"

Jump 3 = opt "3"

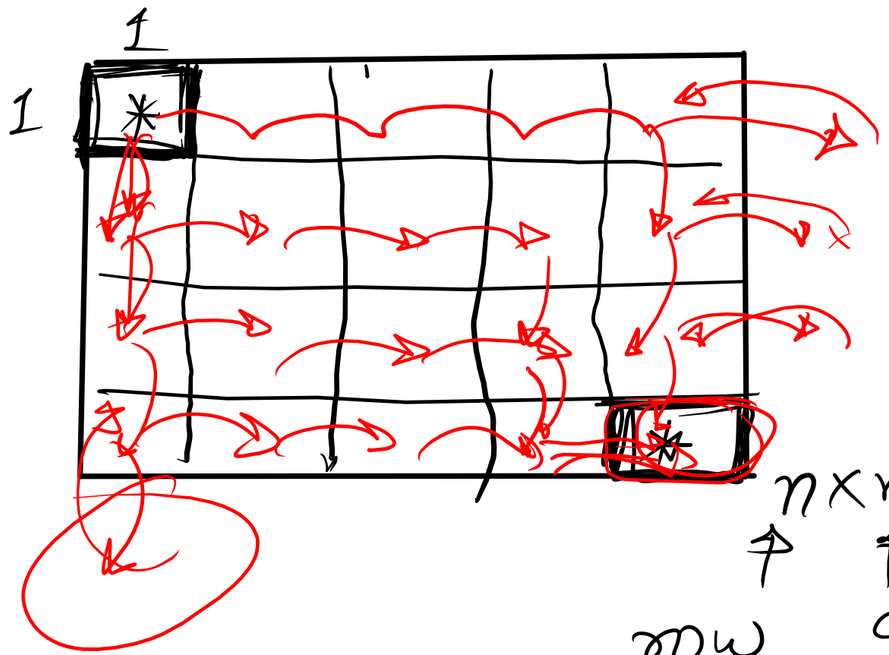


f(3, " ")
3
f(2, "1")
2
f(1, "11")
1
f(0, "111")
0
print
f(0, "12")
print

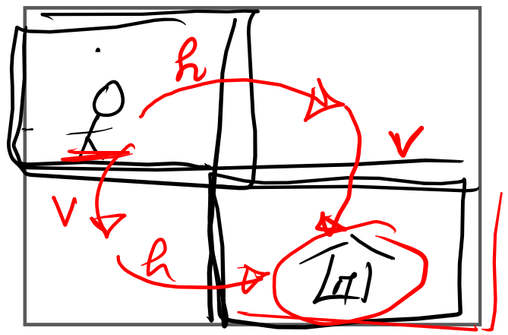
String Palindrome

```java
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);
```

aMaze Problem.



[2, 2]



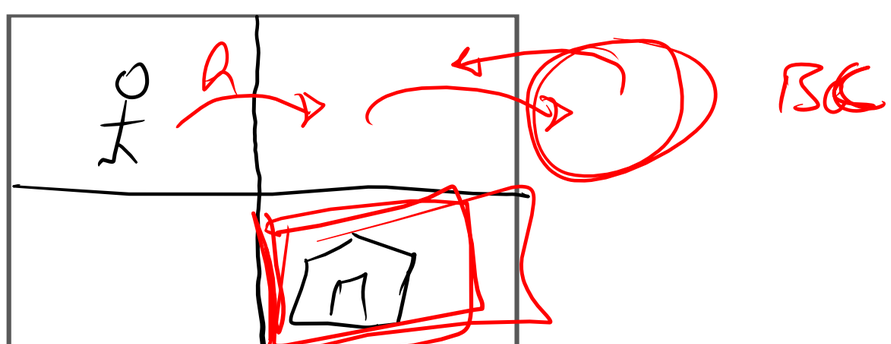
"hv" ✓

"vh" ✓

You can take only 2-moves

- horizontal → 'h'
- Vertical → 'v'

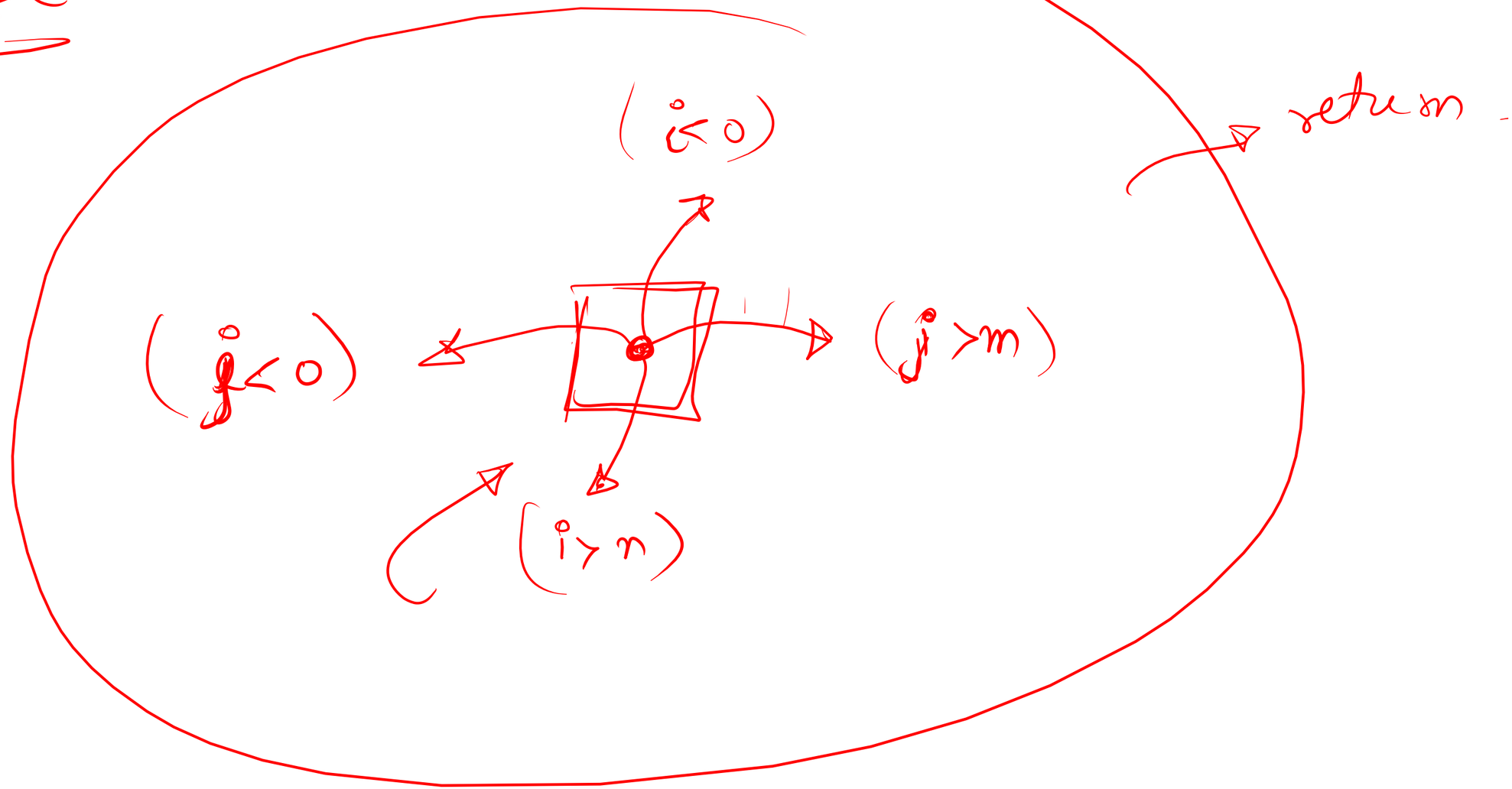
if(i



BC

```
if (i=n && j==m)
    print;
    return;
}
```

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-9ea6-12902858badf

* aMaze Problem

still Doubt?

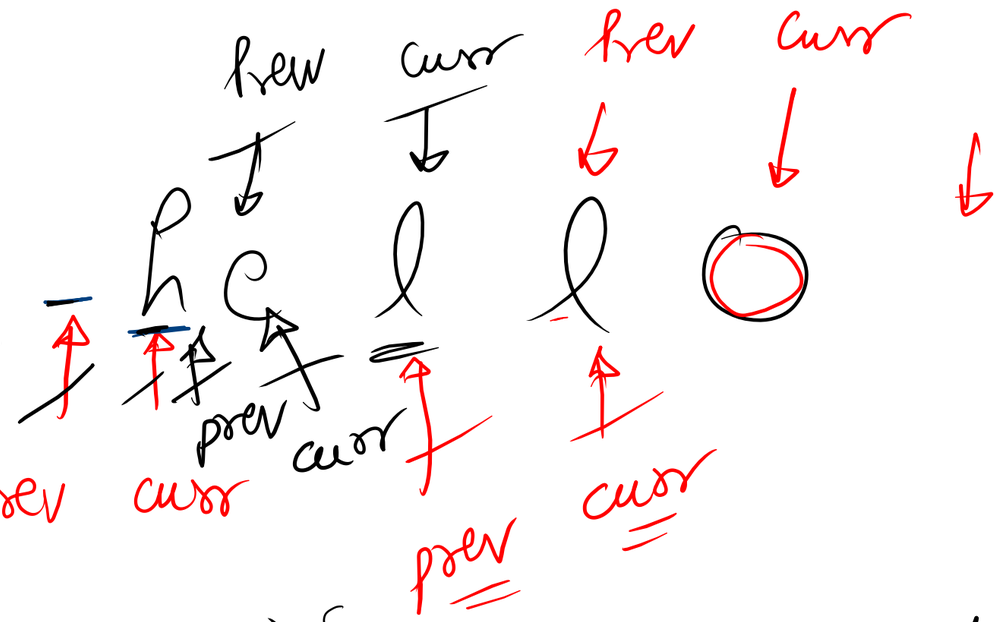
* Pair Star

hello → hel*lo

i/p

o/p

str:



o/p = "hel*lo"

```
if(prev == curr){
    op + '*' + curr
}
else {
    op + curr
}
```

```java
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 ste , output , idx = 0 )

}
```

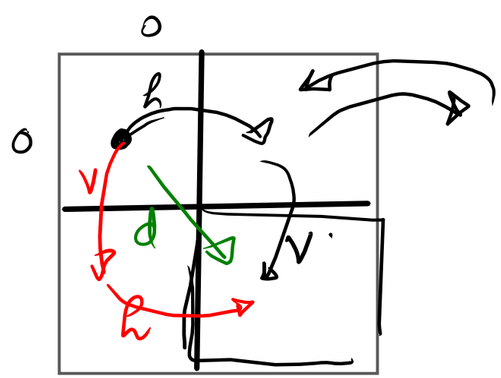
Q. Maze Problem

Input

n = 2
m = 2

Output

```
h1v1
v1h1
d1
```

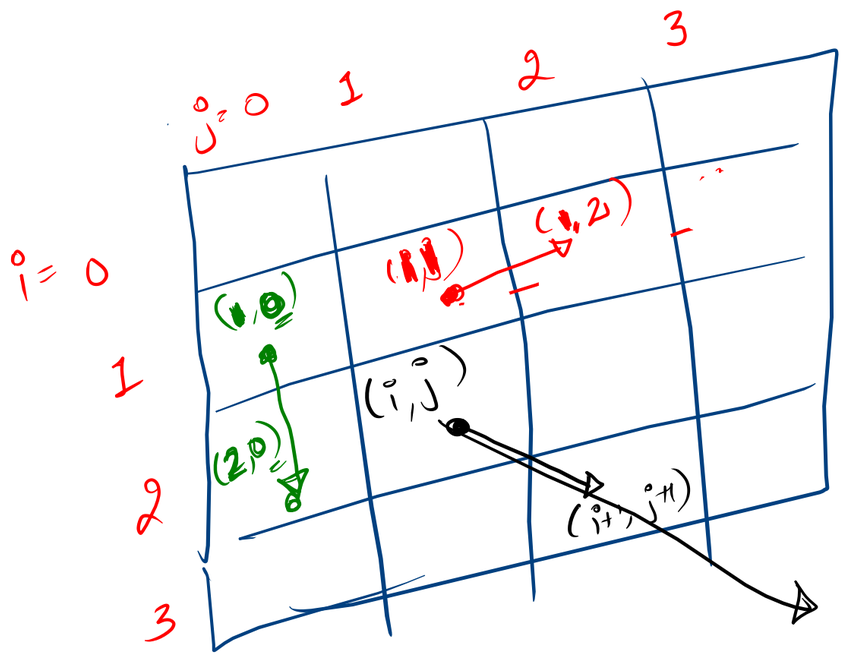


h1v1

v1h1

d1

index game



for Horizontal move

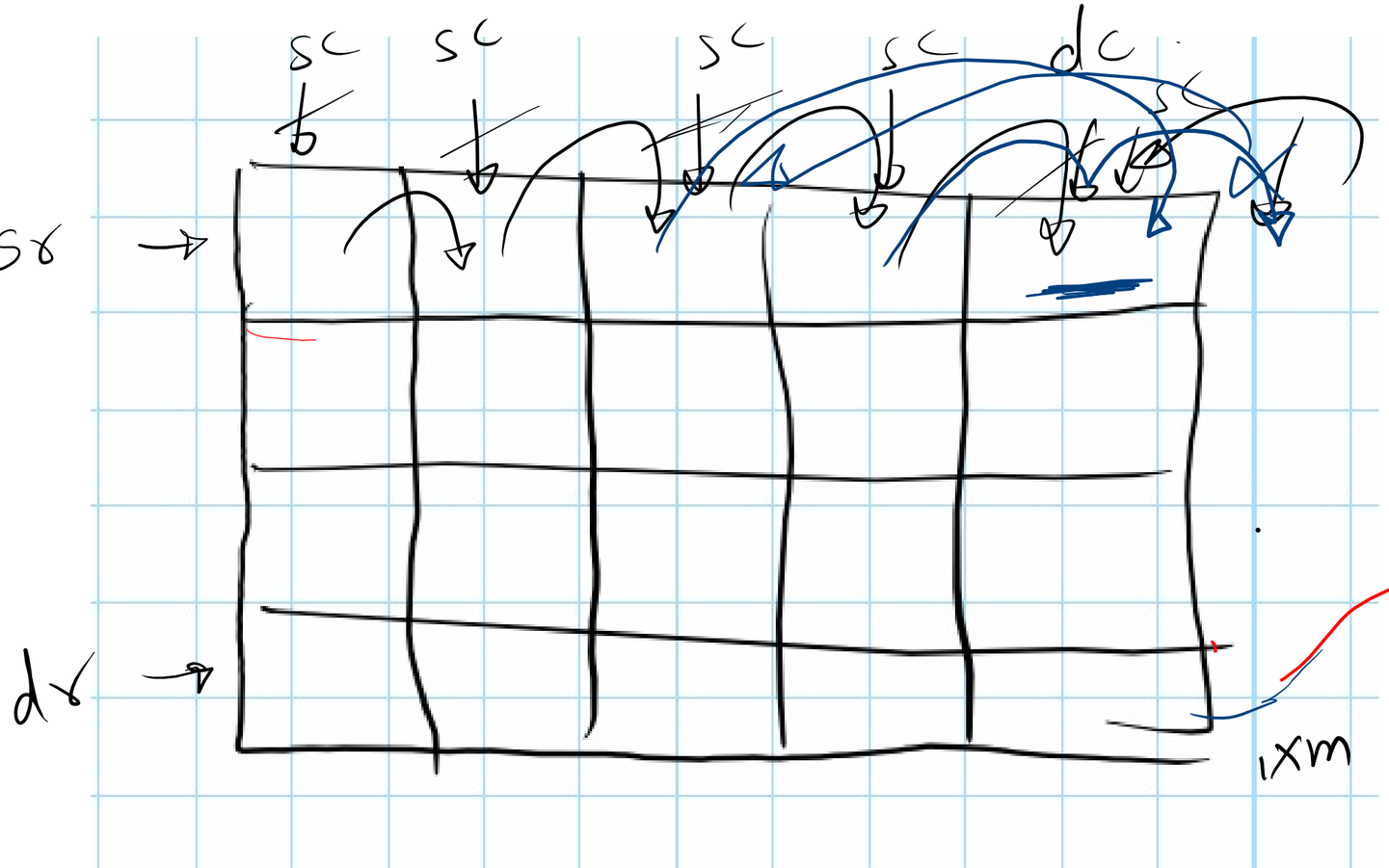
$$(i,j) \rightarrow (i, j+x)$$

for Vertical move

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

for diagonal move

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



for

sc + 2

sc + 3

sc + 4

sc + 5

```
if( sx == n && sc == m){
    print(answer);
    return;
}
```

Base case

Towers of Hanoi