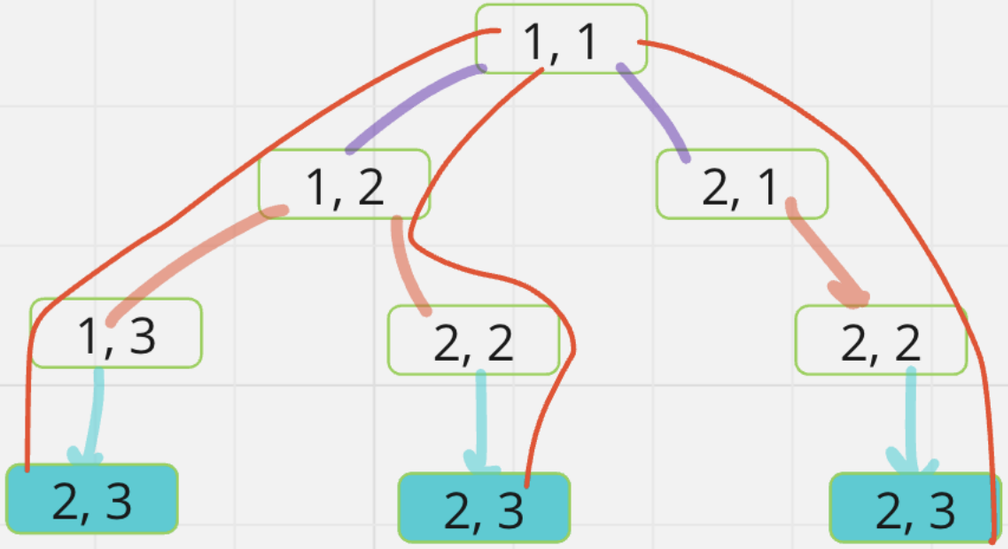
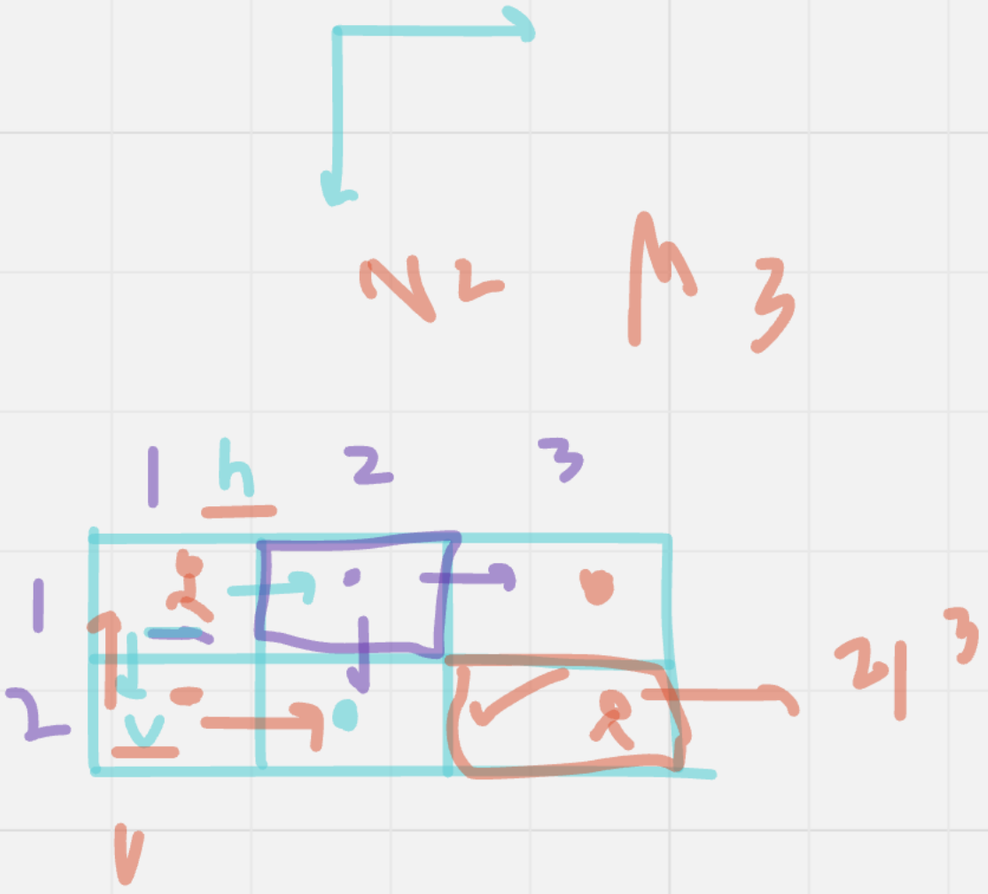
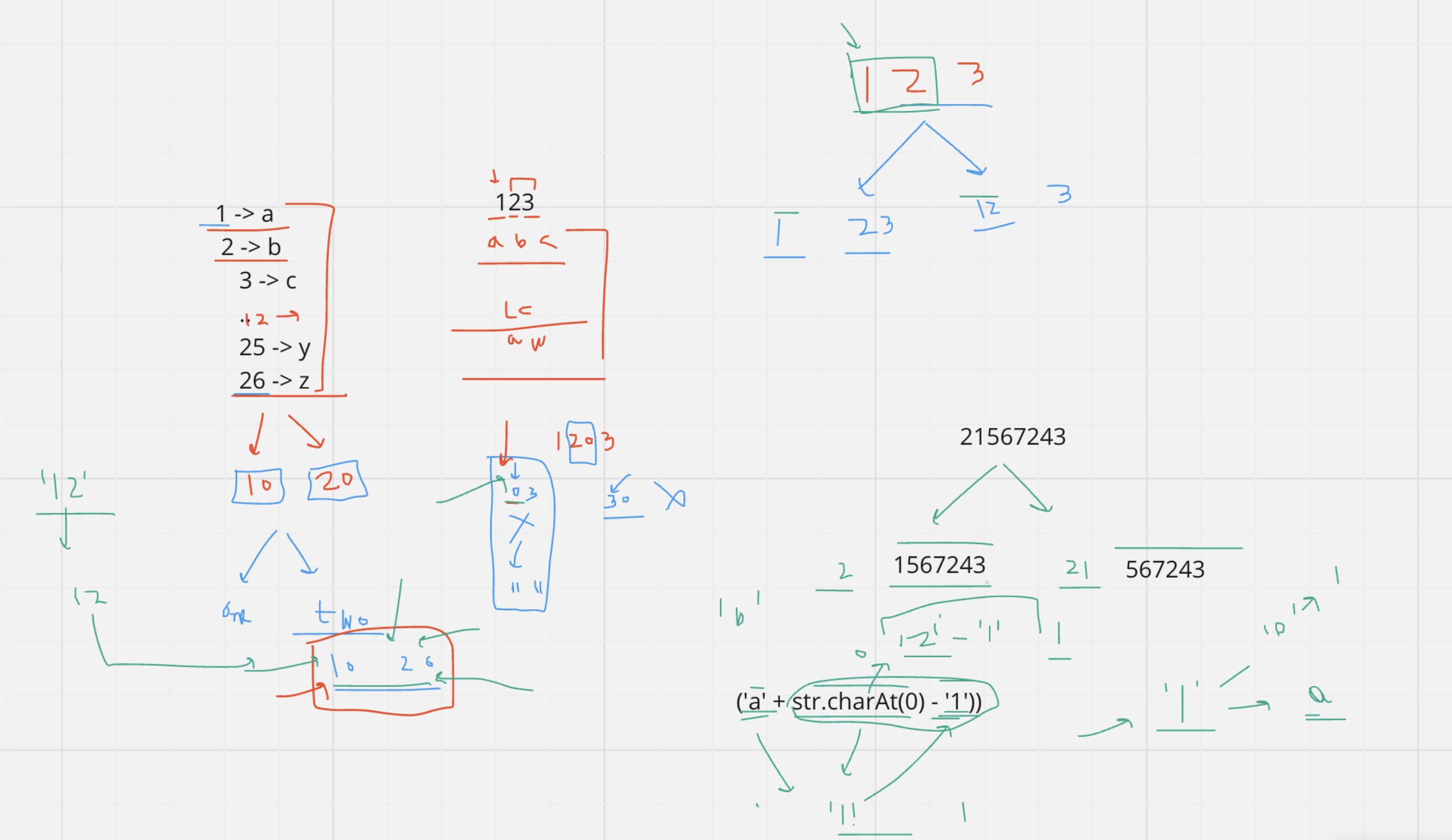

1, 1
1, 2
2, 1
1, 3
2, 2
2, 2
2, 3
2, 3
2, 3





1 -> a
2 -> b
3 -> c
+2 →
25 -> y
26 -> z

123

a b c

Lc

a w

1 2 3

1 23 12 3

10 20

'12'

12

one two

10 26

1 20 3

10 3

11 11

30

21567243

2 1567243 21 567243

'b'

'2' - '1' 1

0

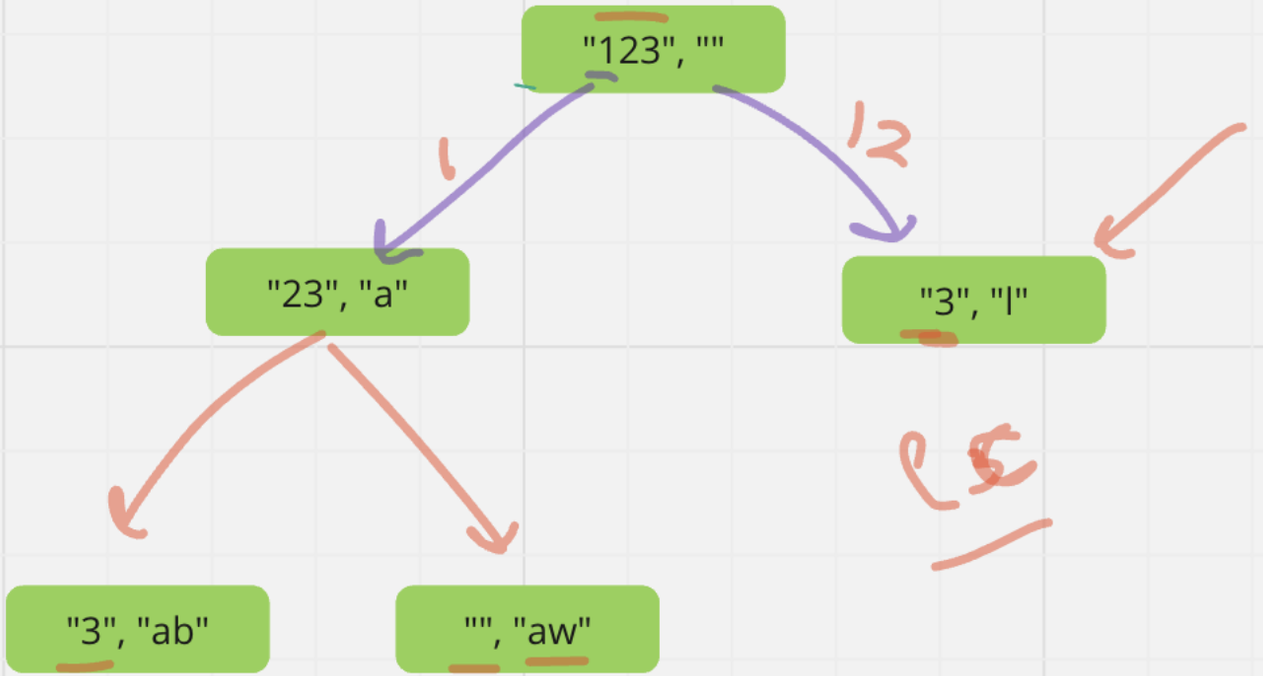
('a' + str.charAt(0) - '1'))

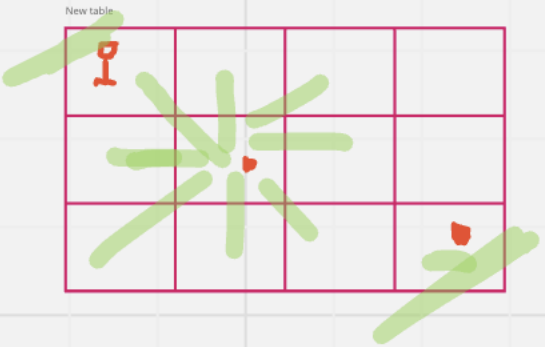
'1' 1

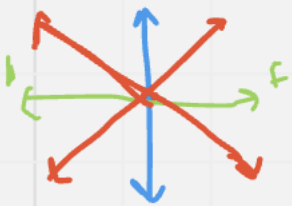
'1' a

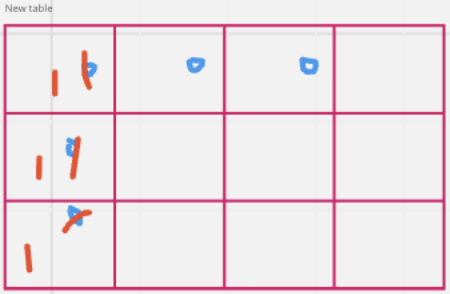
10 1

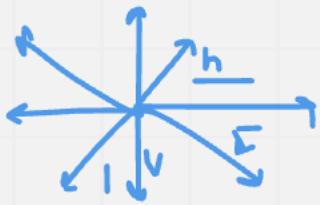
```
static void helper(String str, String ans) {

    if(str.length() == 0) {
        System.out.println(ans);
        return ;
    }

    if(str.length() == 1) {
        System.out.println(ans + (char)('a' + str.charAt(0) - '1'));
        return;
    }

    // safe condi
    if(str.charAt(0) == '0') {
        return ;
    }

    // pick one char
    helper(str.substring(1), ans + (char)('a' + str.charAt(0) - '1'));
    // two char
    // 23
    int val = Integer.parseInt(str.substring(0, 2));
    if(val >= 10 && val <= 26) {
        helper( str.substring(2), ans + (char)('a' + val - 1) );
    }

}
```

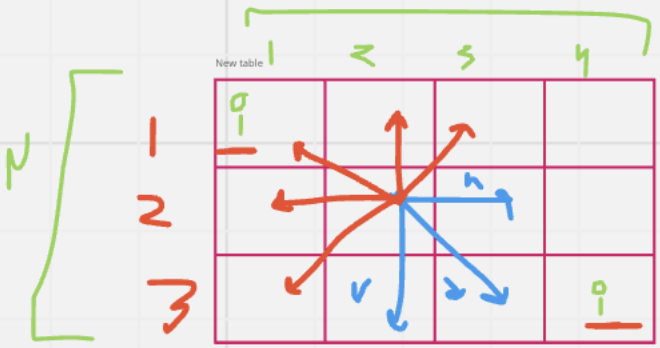



New table

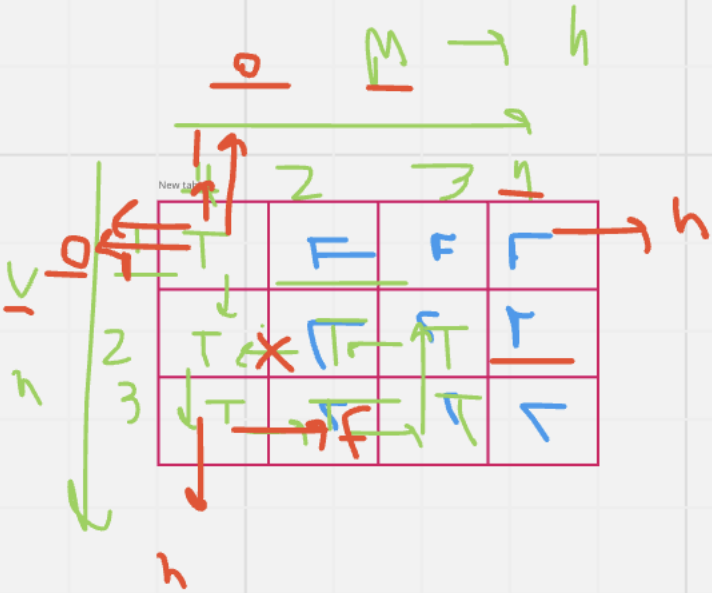


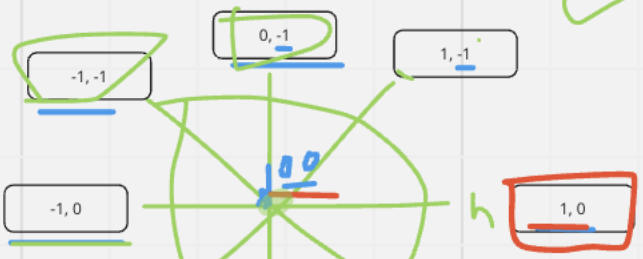

New table



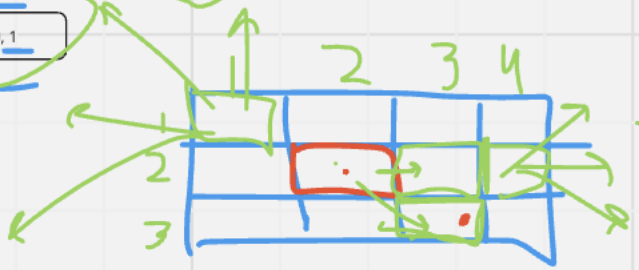

New table

```
// safe condi
    if(h <= 0 || v <= 0 || h > m || v > n) {
      return 0;
    }
```

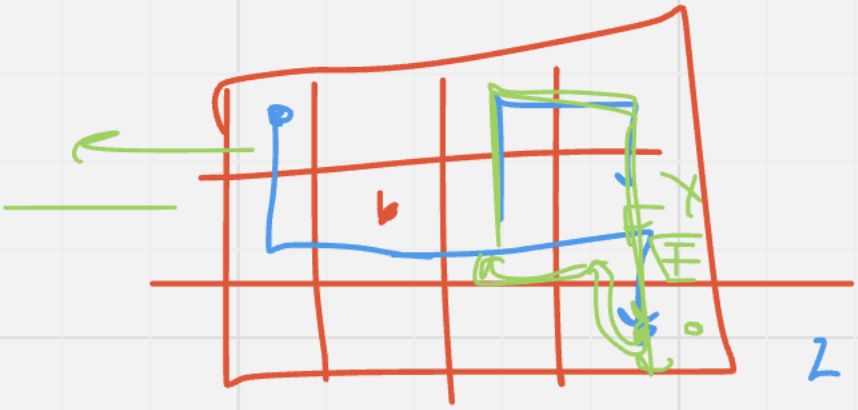




h	1	1	0	-1	-1	-1	0	1
v	0	1	1	1	0	-1	-1	-1



```
int dirH = {1, 1, 0, -1, -1, -1, 0, 1};
    int dirV = {0, 1, 1, 1, 0, -1, -1, -1};

for(int k = 0; k < 8; k++) {
    int nh = h + dirH[k];
    int nv = v + dirV[k];
    if(isSafe(n, m, nh, nv) && !vis[nv][nh]) {
        vis[nv][nh] = true;
        ans += countAllPath(n, m, nh, nv, vis);

    }

}
```



Graph