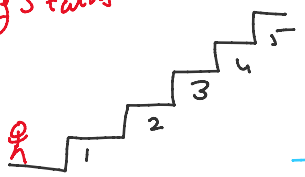
Rec-3

17 December 2022 11:08

Climbing Stairs



1,2,3

1,1,1,1,1

1,1,1,2

1,2,1,1

2,1,1,1

1,1,2,1

2,2,1

1,2,2

2,1,2

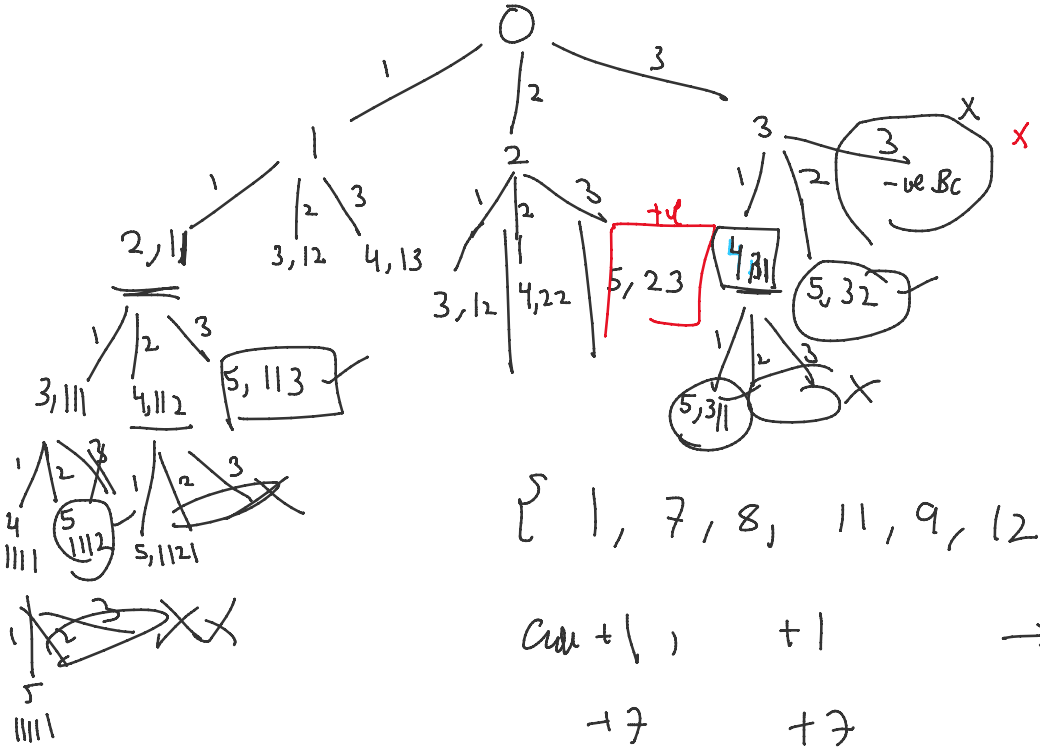
3,1,1

1,3,1

1,1,3

2,3

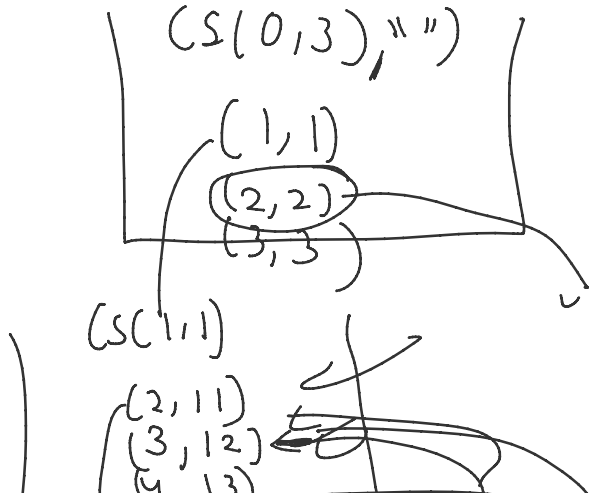
3,2



{ 1, 7, 8, 11, 9, 12 }

Call + 1 , +1 → (Call + call[i] , path + i)

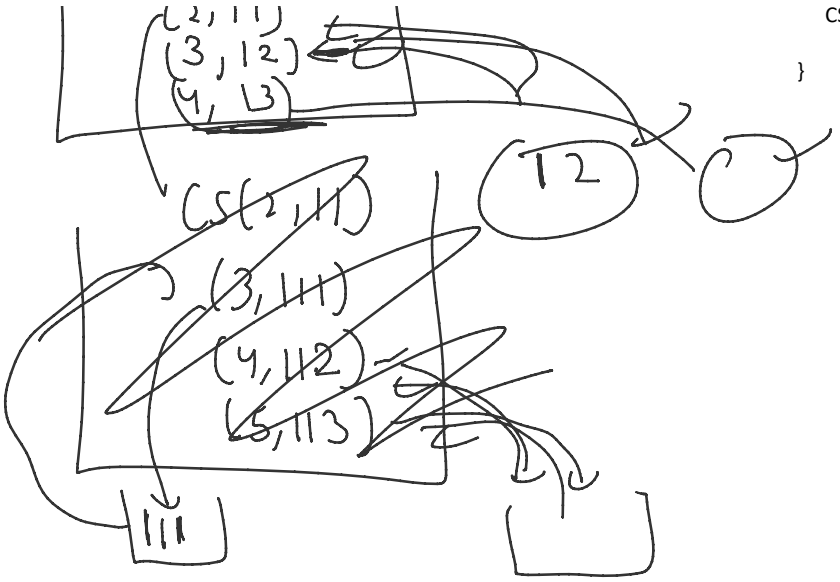
+7 +7




```
public static void main(String[] args) {
        CStairs(0, 3, "");
    }
    public static void CStairs(int curr, int dest, String
    path) {
        if(curr==dest) {// +ve positive BC
            System.out.println(path);
            return;
        }
        if(curr>dest) { //-ve negitive BC
            return;
        }
        CStairs(curr+1, dest, path+1);//1 ki jump
        CStairs(curr+2, dest, path+2);//2 ki jump
        CStairs(curr+3, dest, path+3);//3 ki jump

    }
```

CStairs(curr+3, dest, path+3);//3 ki jump

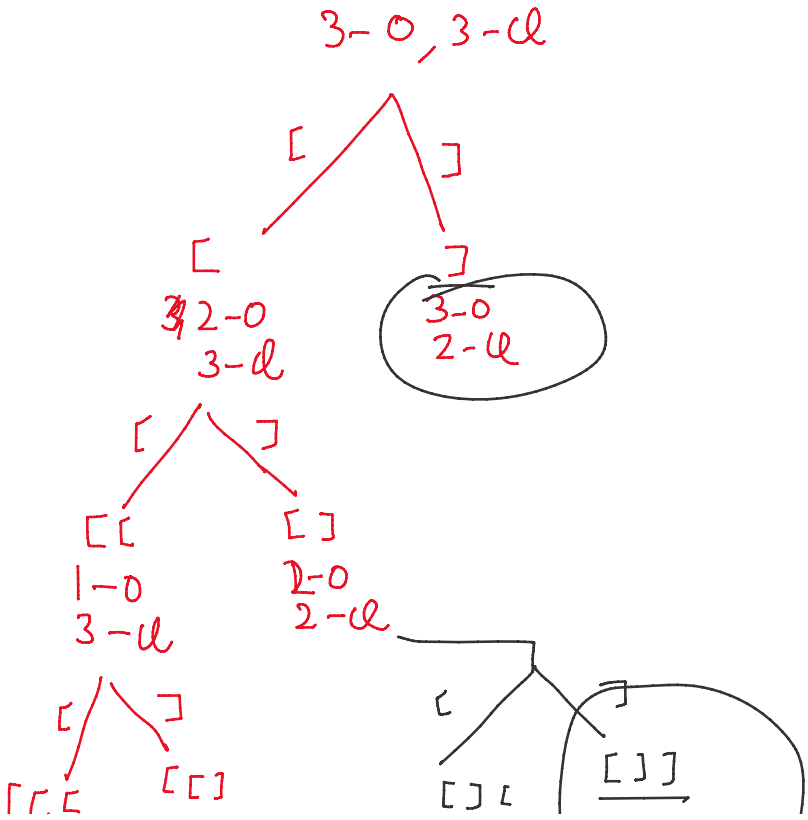
}

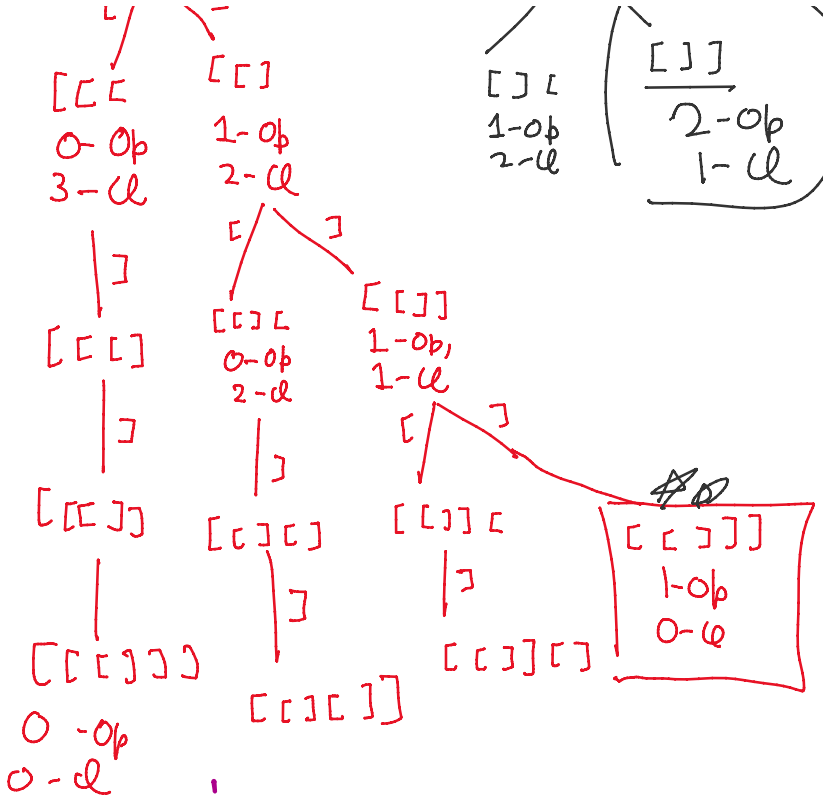


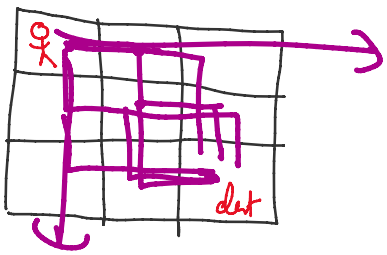
1, 2

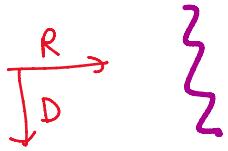
A-

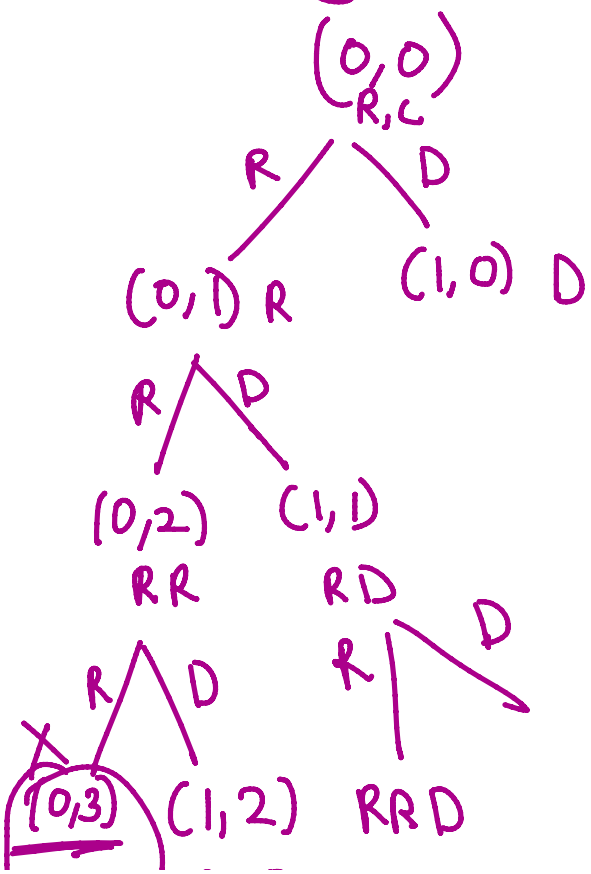
n = 3











RRDD | ~~DRRD~~

RDRD | DRDR

RDDR | DDRR

(0,3) RR

(1,2) RRD

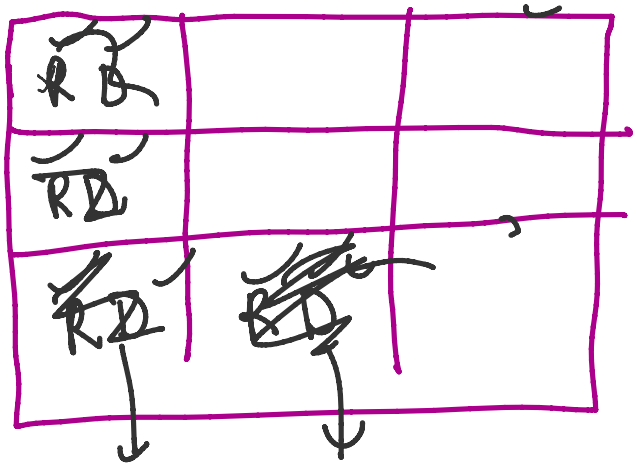
RRD

R / D

(1,3) X

R RDD

(2,2).



DDRR

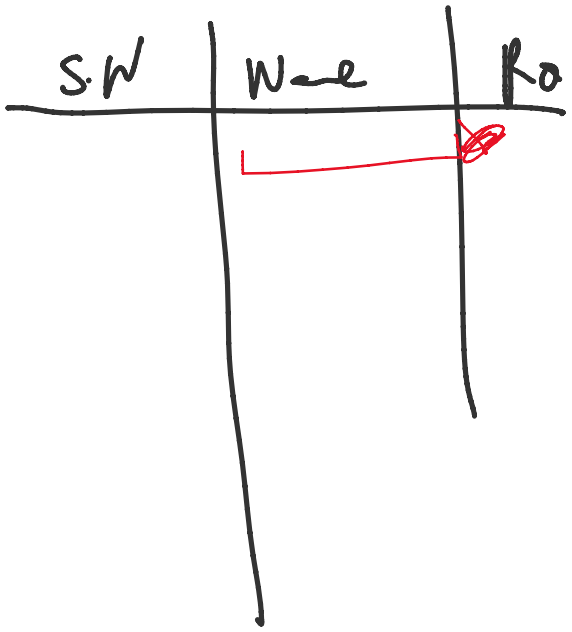
RRDD

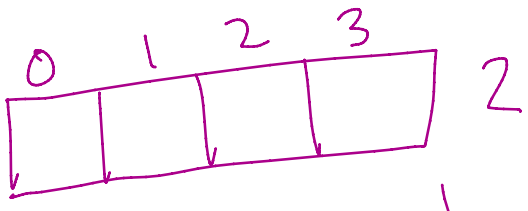
RDRD

RDDR

DRRD

DRDR

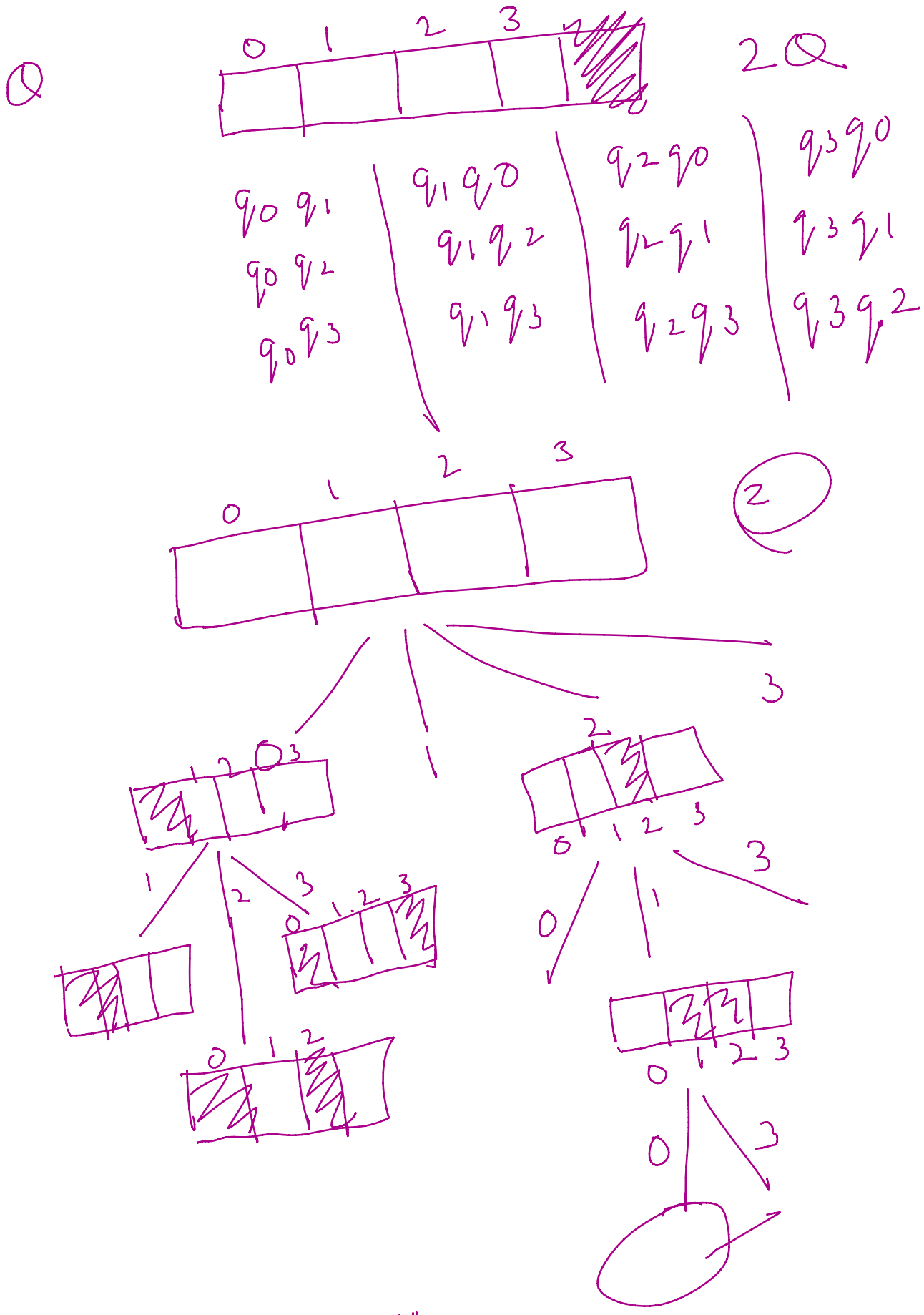




$q_0 q_1$ $q_0 q_2$ $q_0 q_3$

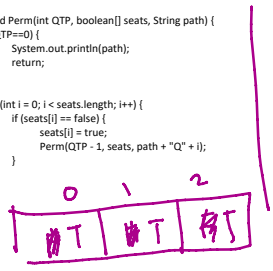
$q_1 q_2$ $q_1 q_3$

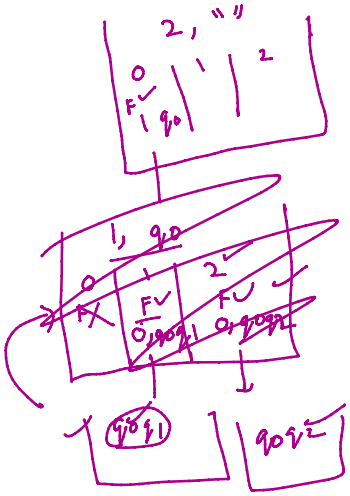
$q_2 q_3$



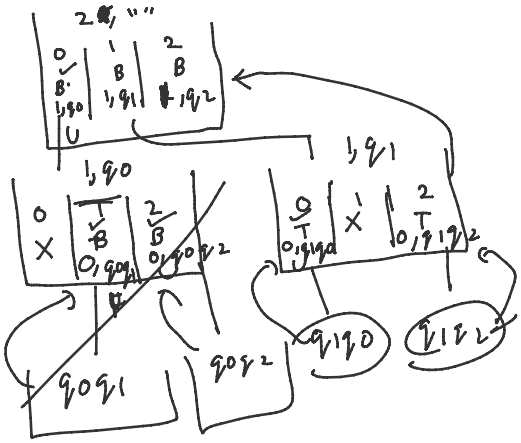

```
public static void Perm(int QTP, boolean[] seats, String path) {
        if(QTP==0) {
            System.out.println(path);
```

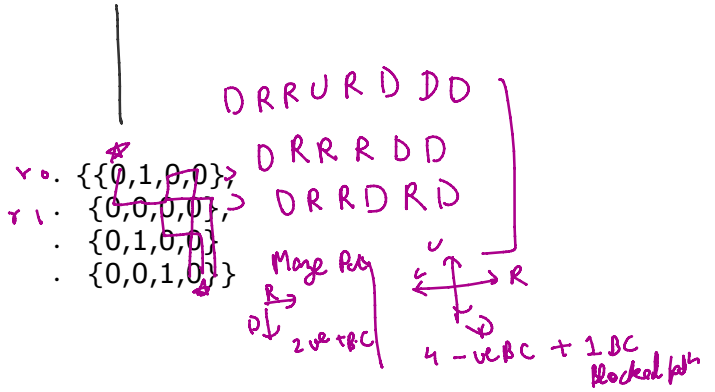
```
public static void Perm(int QTP, boolean[] seats, String path) {
        if(QTP==0) {
                System.out.println(path);
                return;

        }
        for (int i = 0; i < seats.length; i++) {
                if (seats[i] == false) {
                        seats[i] = true;
                        Perm(QTP - 1, seats, path + "Q" + i);
                }
        }
}
```

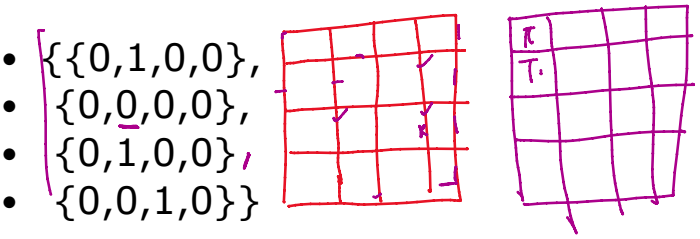




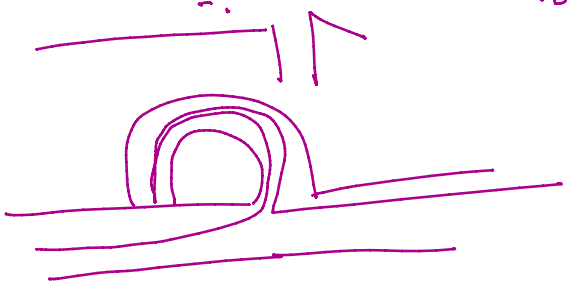
```
public static void Perm(int QTP, boolean[] seats, String path) {
        if(QTP==0) {
                System.out.println(path);
                return;

        }
        for (int i = 0; i < seats.length; i++) {
                if (seats[i] == false) {
                        seats[i] = true;
                        Perm(QTP - 1, seats, path + "Q" + i);
                        seats[i] = false;
                }
        }
}
```



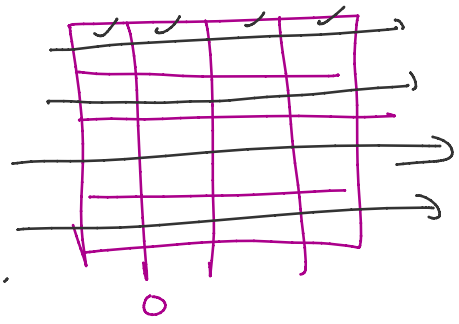




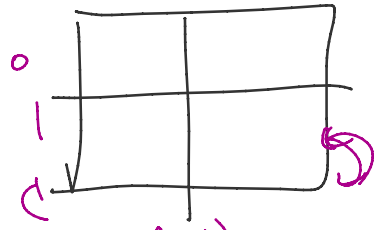
U R K D R D



N - Queens || combinatorial



0



0

1

$q(1,0)\ q(1,1)$

(r,c)

$q(0,0)\ (0,1)$

$q(0,0)\ q(1,0)$

$q(0,0)\ q(1,1)$

$q(0,0)$, $q(1,0)$

$q(0,1)$, $q(1,1)$