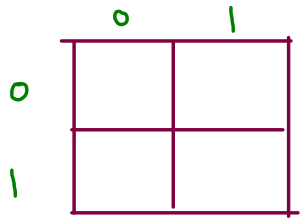
# queen - combination 1 (boxes choose)




N -> order

n -> N * N (boxes)

r -> N (queen)

$^4C_2$ -> 6

q q - -

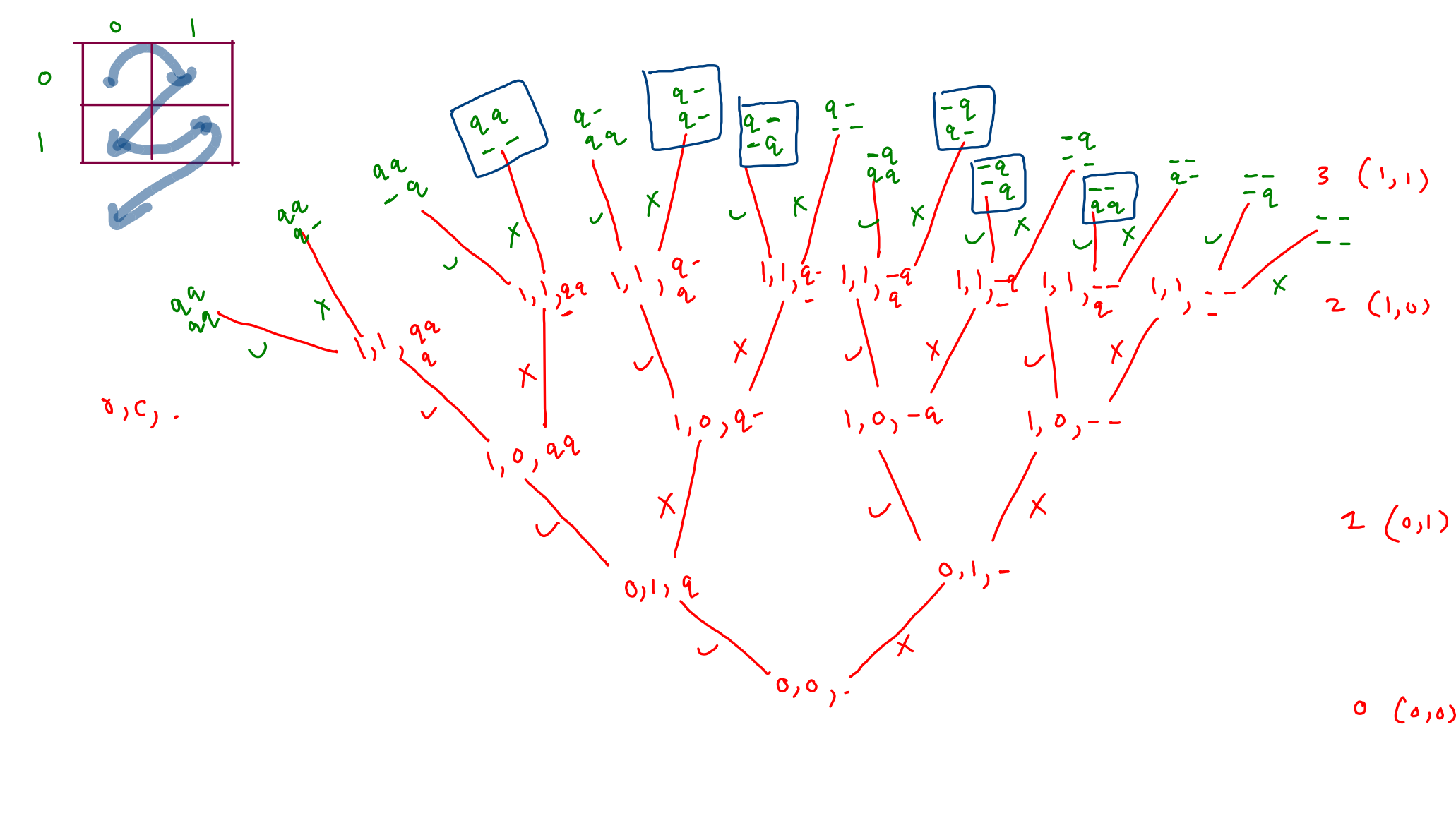
- q q -

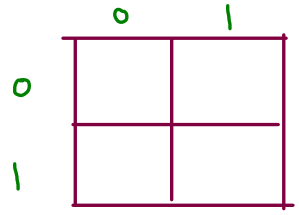
q - q -

q - - q

q - - -

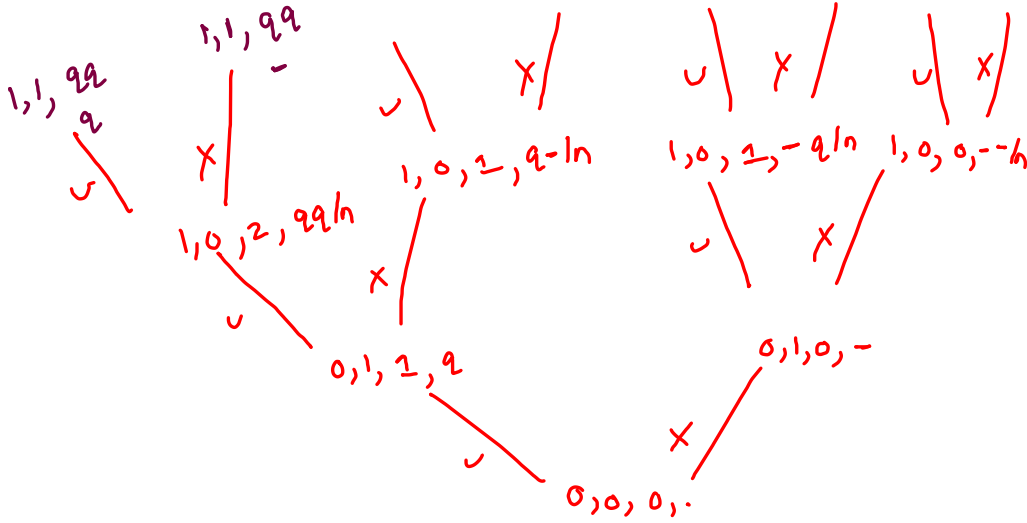
- q q q


0
1
0
1
3 (1,1)
2 (1,0)
2 (0,1)
0 (0,0)




r, c, qpsf, asf

```java
public static void queensCombinations(int qpsf, int N, int row, int col, String asf){
    if(row == N) {
        if(qpsf == N) {
            System.out.println(asf);
        }
        return;
    }

    //yes
    if(col < N-1) {
        queensCombinations(qpsf+1,N,row,col+1,asf + "q");
    }
    else {
        queensCombinations(qpsf+1,N,row+1,0,asf + "q\n");
    }


    //no
    if(col < N-1) {
        queensCombinations(qpsf,N,row,col+1,asf + "-");
    }
    else {
        queensCombinations(qpsf,N,row+1,0,asf + "-\n");
    }
}
```

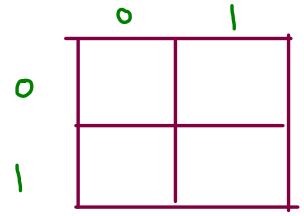


Queen permutations - 1 ( queen chooses)

$n \rightarrow 4$

$r \rightarrow 2$

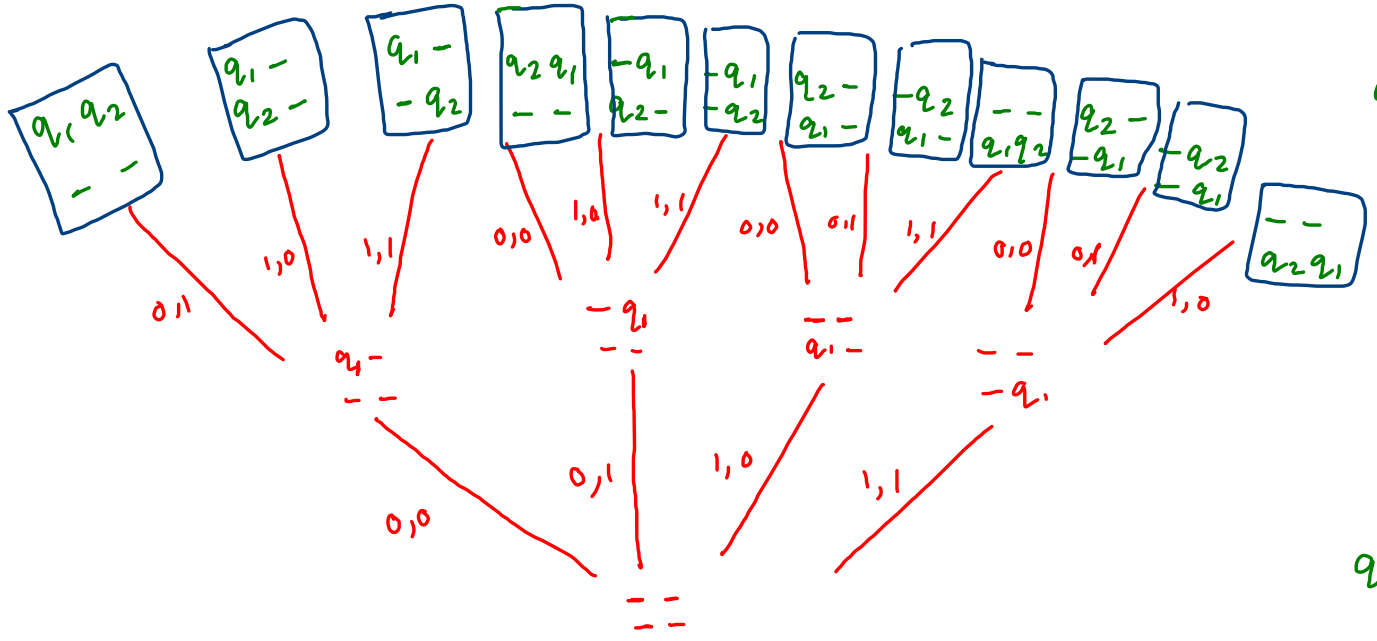
$^4P_2 = 12$

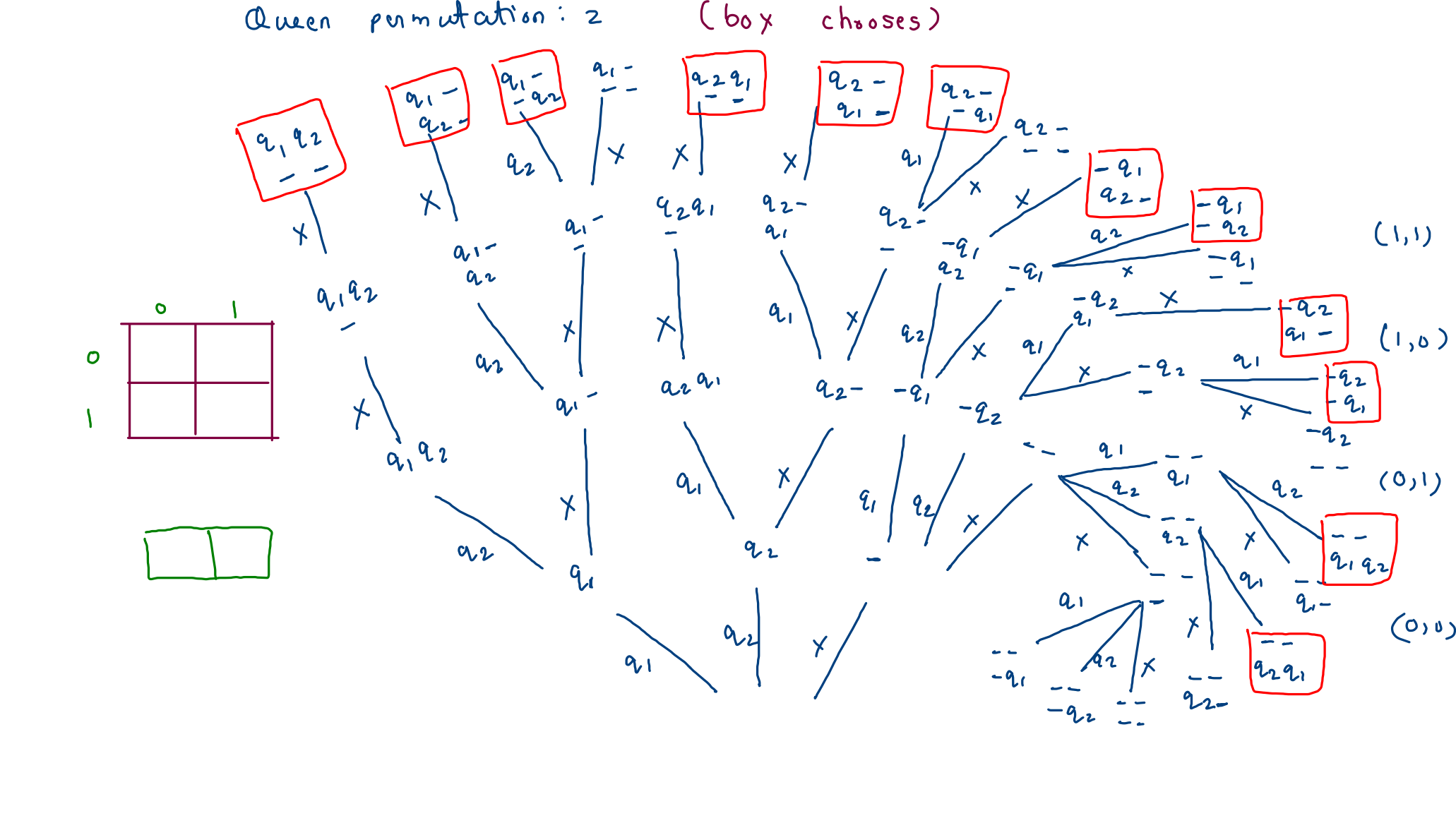



$N \rightarrow 2$

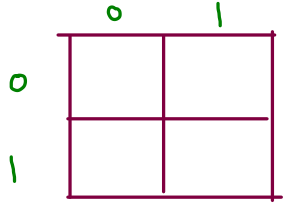
$n \rightarrow N * N$

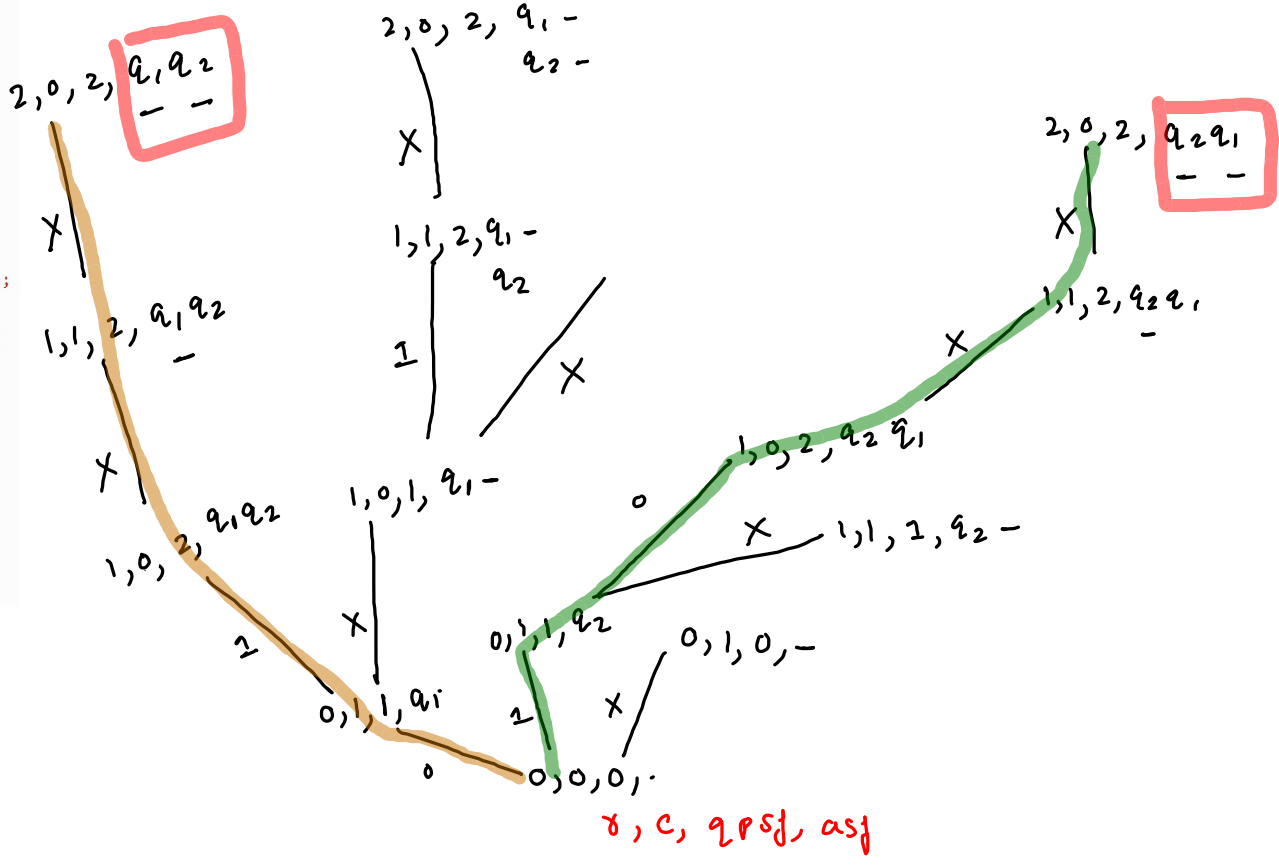
$r = N$



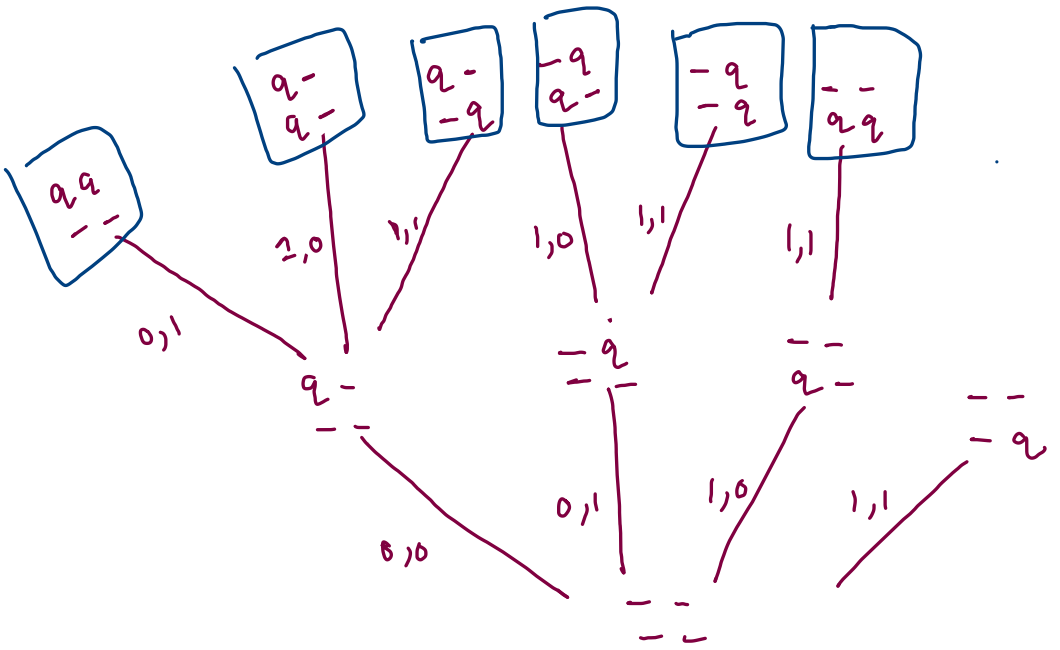

Queen permutation : 2 (box chooses)
0
1
0
1
(1,1)
(1,0)
(0,1)
(0,0)

```java
if(row == N) {
    if(qpsf == N) {
        System.out.println(asf);
        System.out.println();
    }

    return;
}

//box choice -> yes calls
for(int q = 0; q < N;q++) {
    if(queens[q] == false) {
        queens[q] = true;
        if(col < N-1) {
            queensPermutations(qpsf+1,N,row,col+1,asf + "q" + (q+1) + "\t",queens);
        }
        else {
            queensPermutations(qpsf+1,N,row+1,0,asf + "q" + (q+1) + "\n",queens);
        }
        queens[q] = false;
    }
}

//box no call
if(col < N-1) {
    queensPermutations(qpsf,N,row,col+1,asf + "-\t",queens);
}
else {
    queensPermutations(qpsf,N,row+1,0,asf + "-\n",queens);
}
```



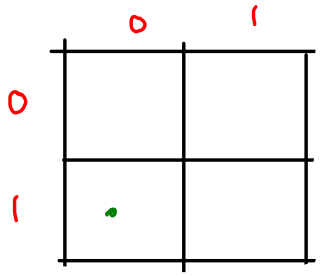



combinations — 2 (queen choice)




q

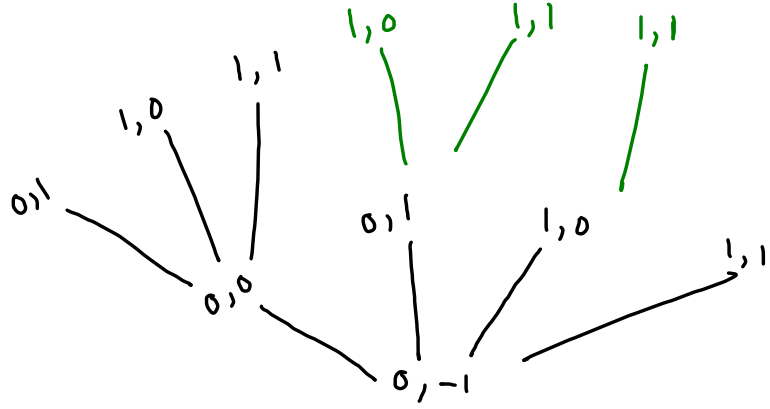
q




dr => last placed queen's row

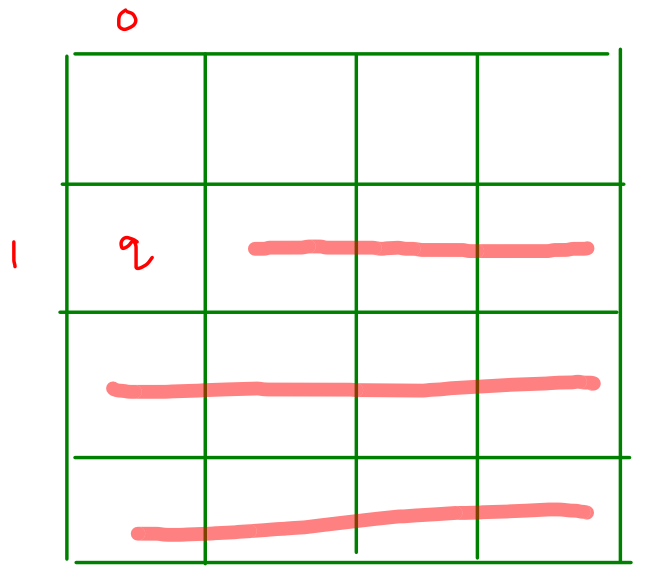
dc => last placed queen cols

```
for (int i = dr ; i<N; i++) {
    for (int j = (i==dr) ? dc+1 : 0 ; j<N; j++)
```



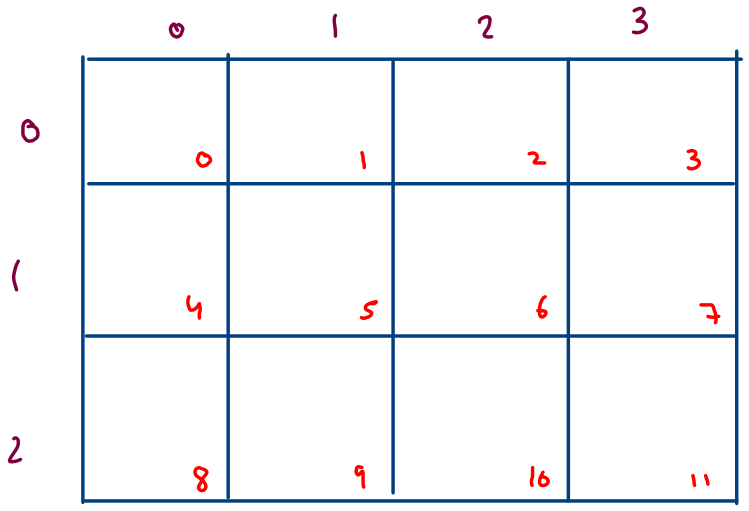

q

```
for (int i = dr ; i < N; i++) {
    for (int j = (i == dr) ? dc+1 : 0 ; j<N;j++)
```




$d_r = 1$

$d_c = 0$




(2d to 1d)

i, j to bno

bno = i * cols + j

(1d to 2d)

bno. to i, j

i = bno. / cols

j = bno.. /. cols

Queen comb 2d as 1d

( queen chooses)