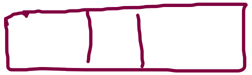
P1

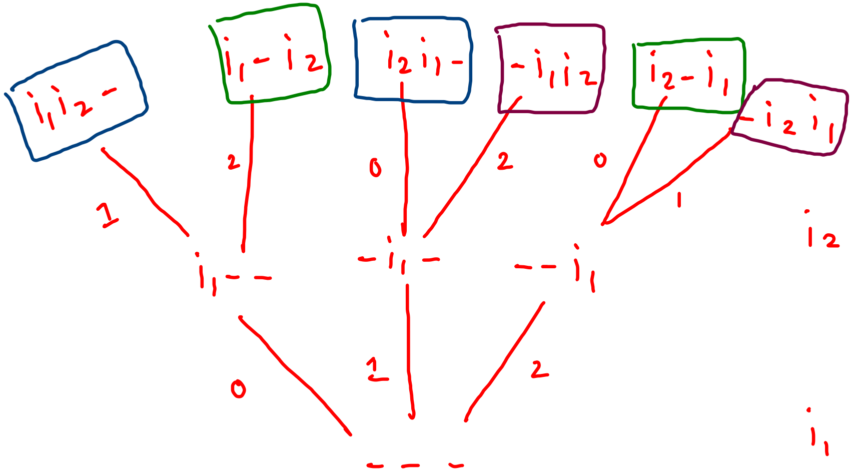
P2 (with the help of c1)

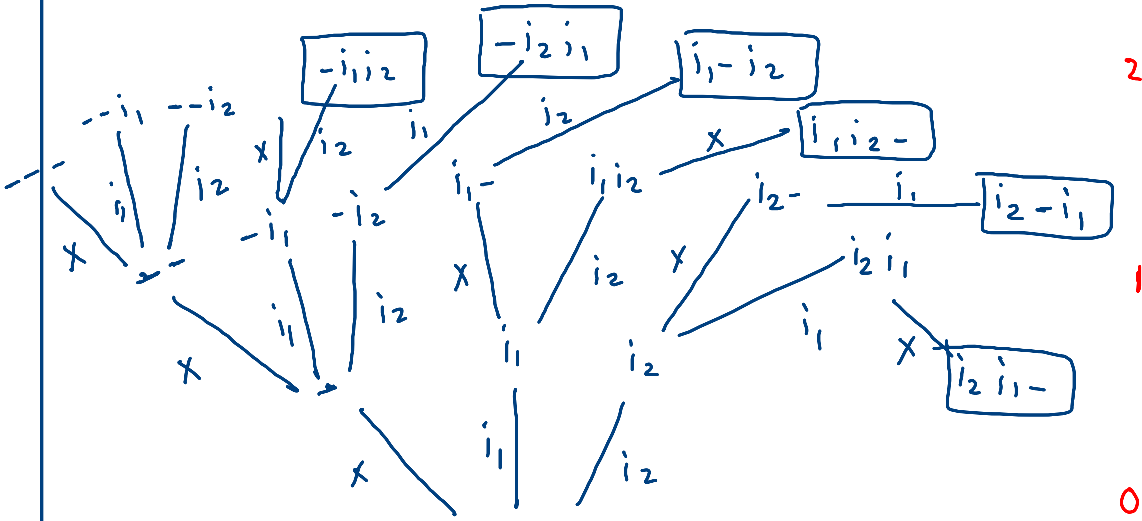
item chooses -> box

n = 3

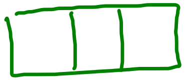
r = 2







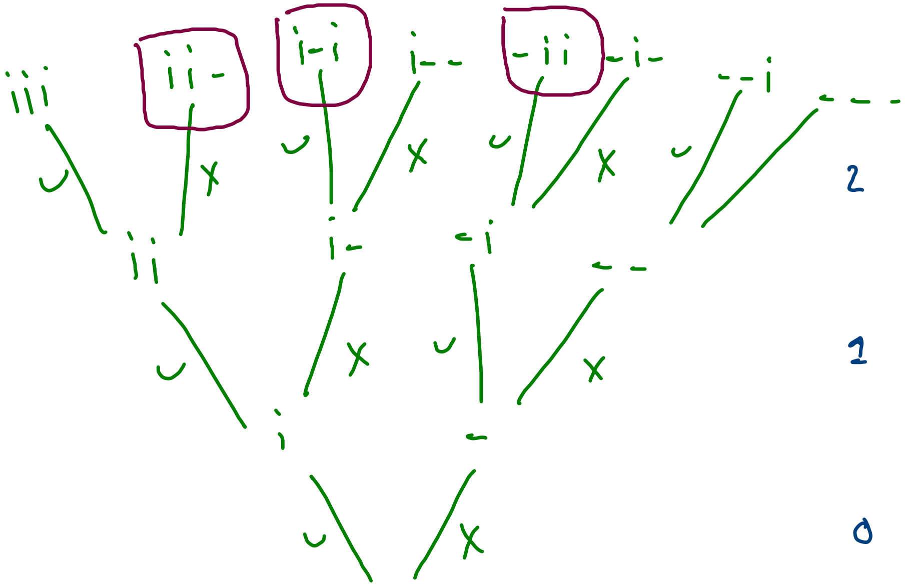
C1



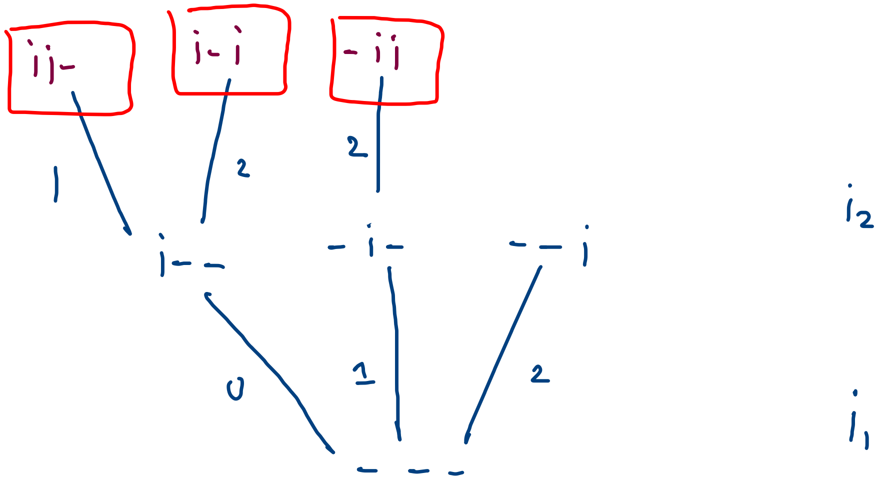
$n = 3$

$r = 2$

$$2^n = {}^nC_0 + {}^nC_1 + {}^nC_2 + \dots {}^nC_n$$



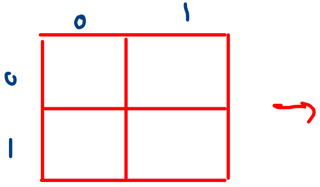
C2 (with the help of P1)



Queens Combinations - 2d As 2d - Box Chooses

level : box

options: yes or no



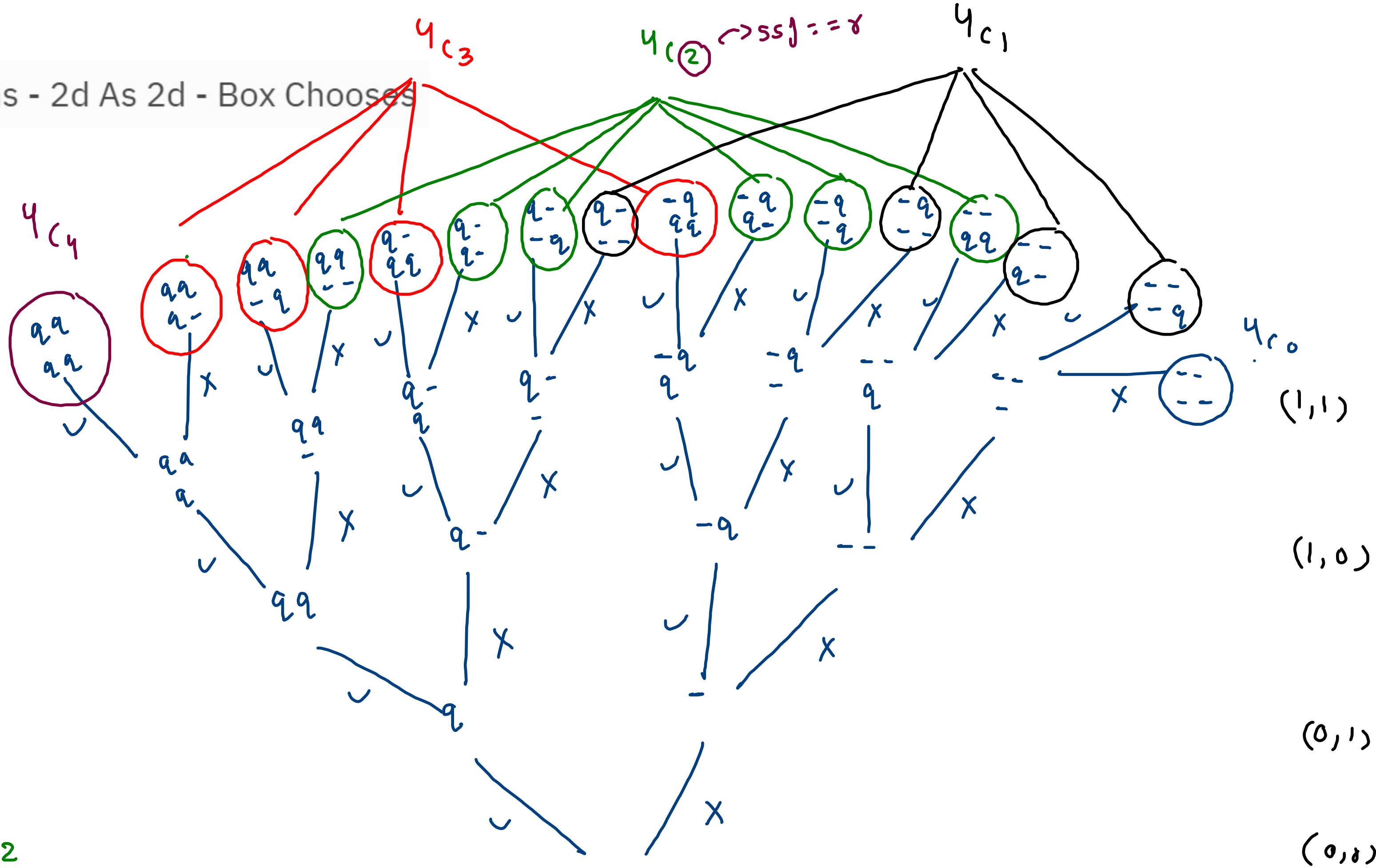
$^{n*n}C_n$

n = 2

total boxes = n*n

= 4

total queen = n = 2



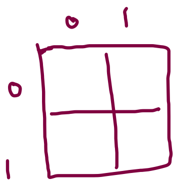
Queens Permutations - 2d As 2d - Queen Chooses

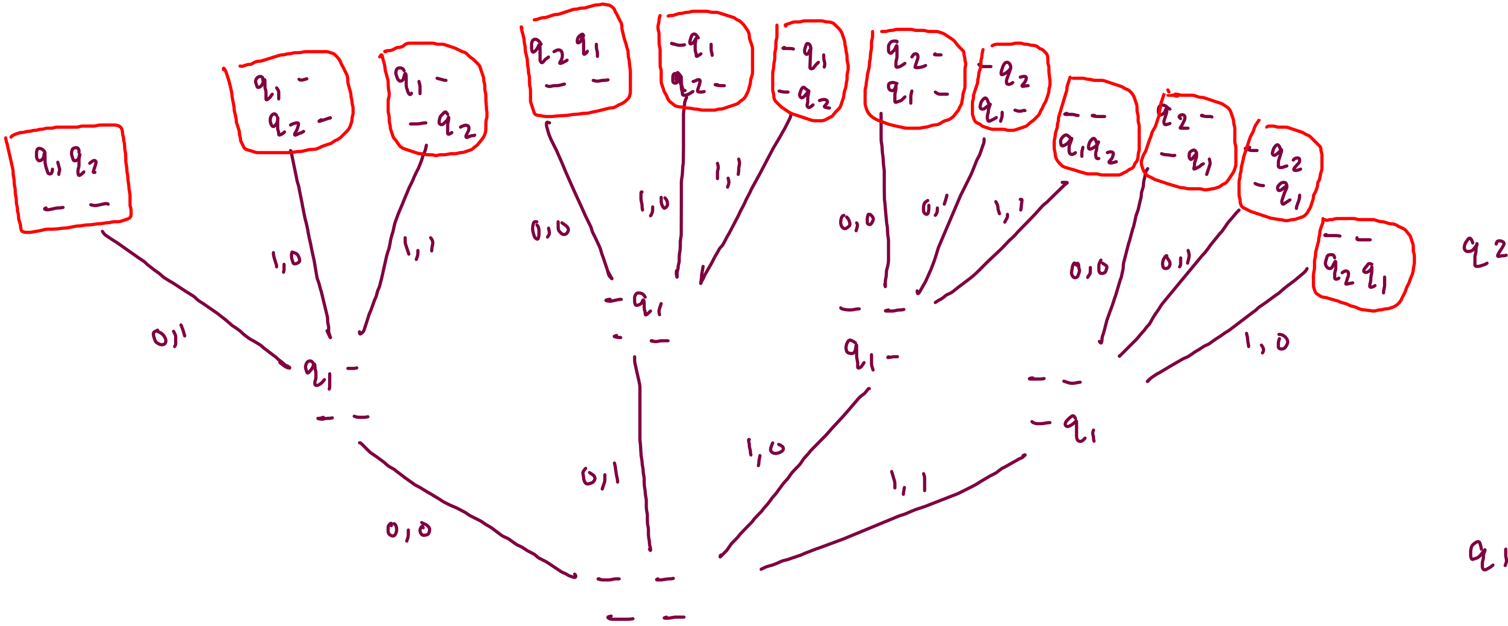
$${}^{4}P_{2} = \frac{4!}{2!} = 12$$

level : queen

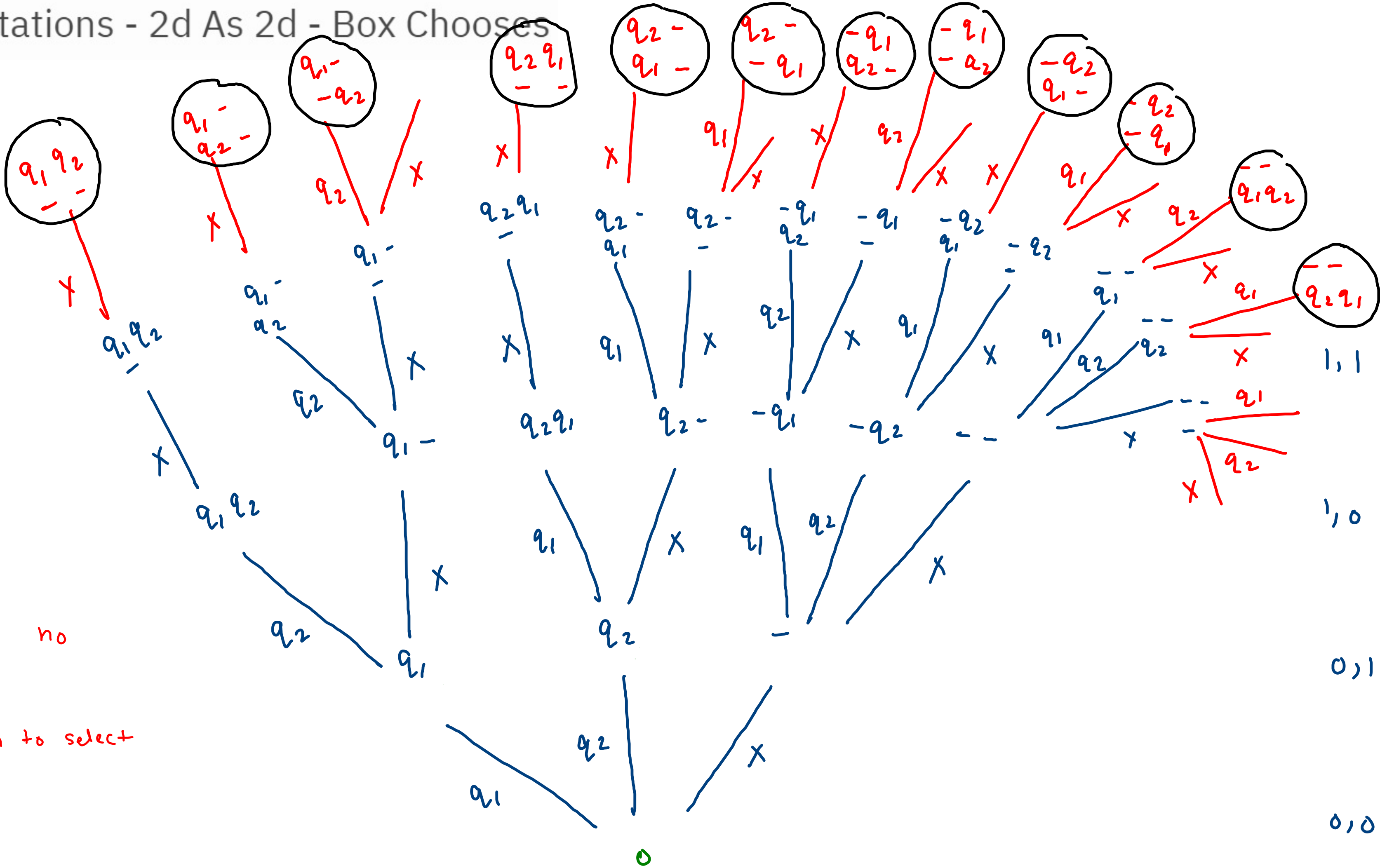
options : boxes

n = 2





Queens Permutations - 2d As 2d - Box Chooses



level - boxes

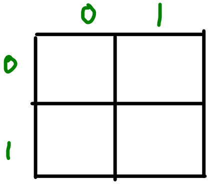
option - yes & no

yes → which queen to select

	0	1
0	1	
1		

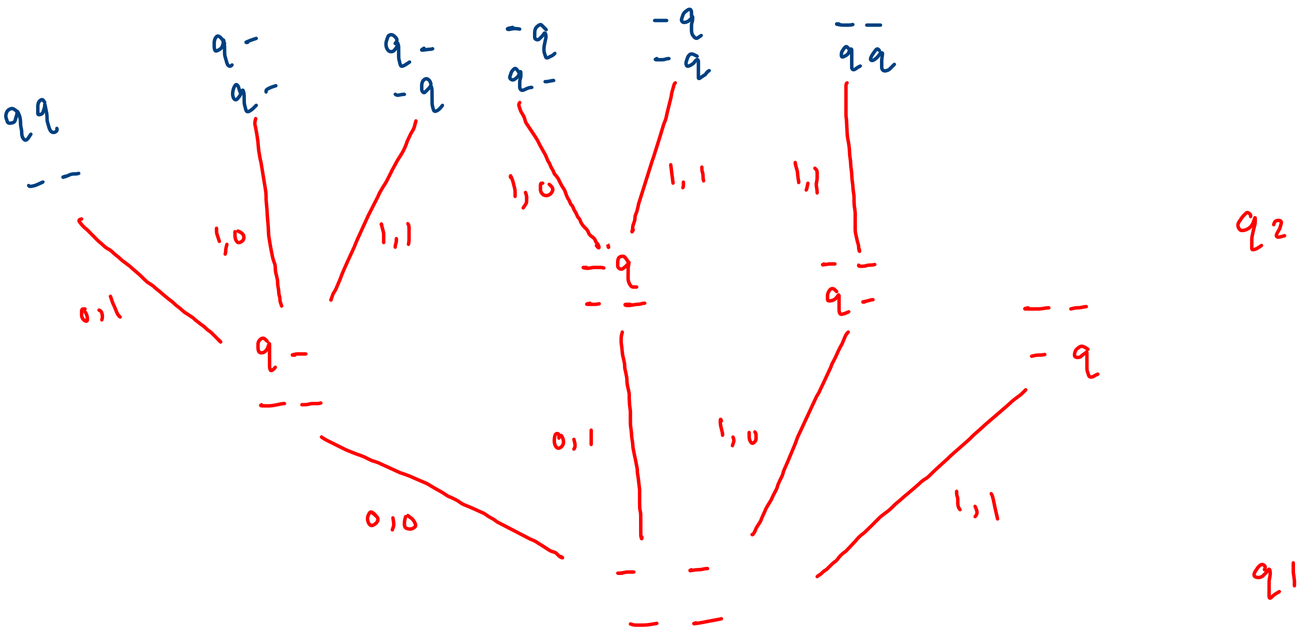
Queens Combinations - 2d As 2d - Queen Chooses

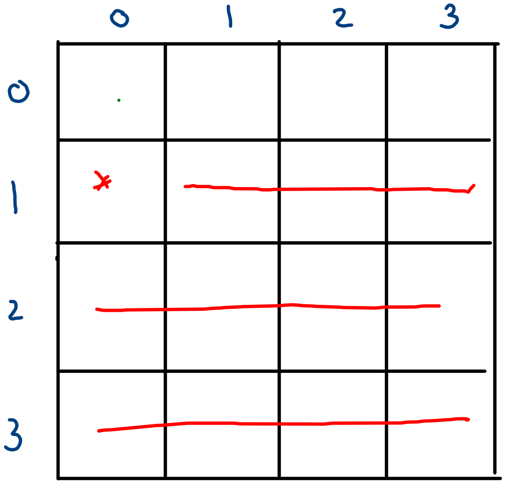
$^4C_2$



level : queen

options : boxes





di = 1

dj = 0

```
for ( int i = di ; i < n; i++ ) {
    for ( j = (i == di) ? dj+1 : 0 ; j < n ; j++) {
    }
}
```

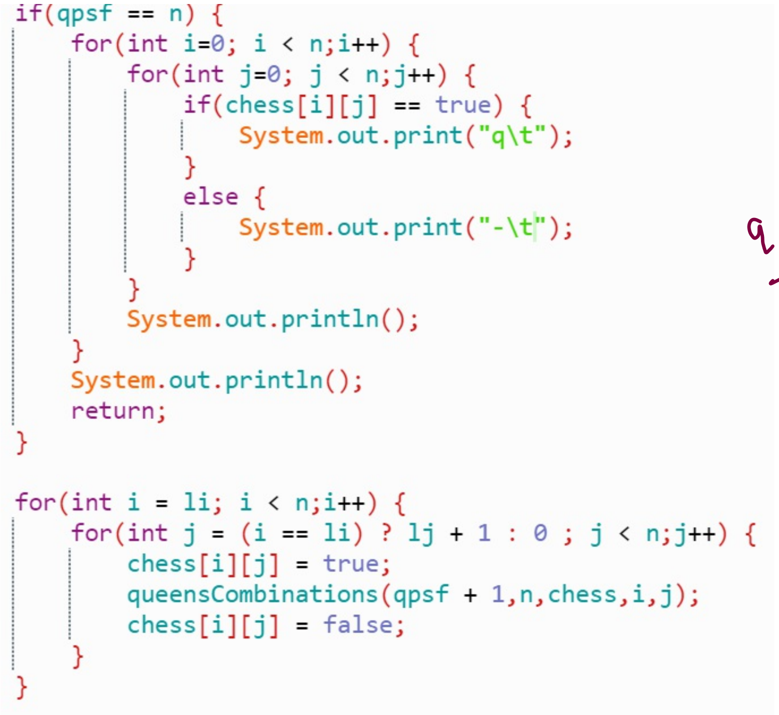
di = 1, dj = 1

i = 1, j = 2, 3

i = 2 j = 0, 1, 2, 3

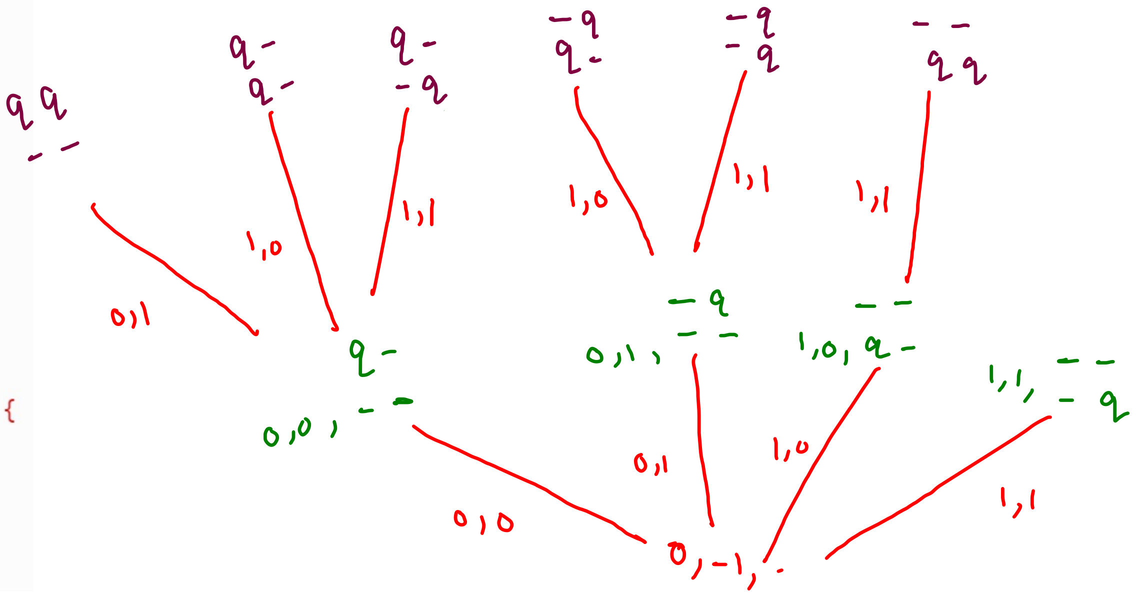
i = 3 j = 0, 1, 2, 3

	0	1
0		
1		



```
if(qpsf == n) {
    for(int i=0; i < n;i++) {
        for(int j=0; j < n;j++) {
            if(chess[i][j] == true) {
                System.out.print("q\t");
            }
            else {
                System.out.print("-\t");
            }
        }
        System.out.println();
    }
    System.out.println();
    return;
}

for(int i = li; i < n;i++) {
    for(int j = (i == li) ? lj + 1 : 0 ; j < n;j++) {
        chess[i][j] = true;
        queensCombinations(qpsf + 1,n,chess,i,j);
        chess[i][j] = false;
    }
}
```



0,-1,-
0,0
0,1
1,0
1,1
0,0, q - / - -
0,1, - q / - -
1,0, - - / q -
1,1, - - / - q
0,1
1,0
1,1
1,0
1,1
1,1
q q / - -
q - / q -
q - / - q
- q / q -
- q / - q
- - / q q