target sum subset/subseq.

arr → {10, 20, 30, 40, 50}

target → 60

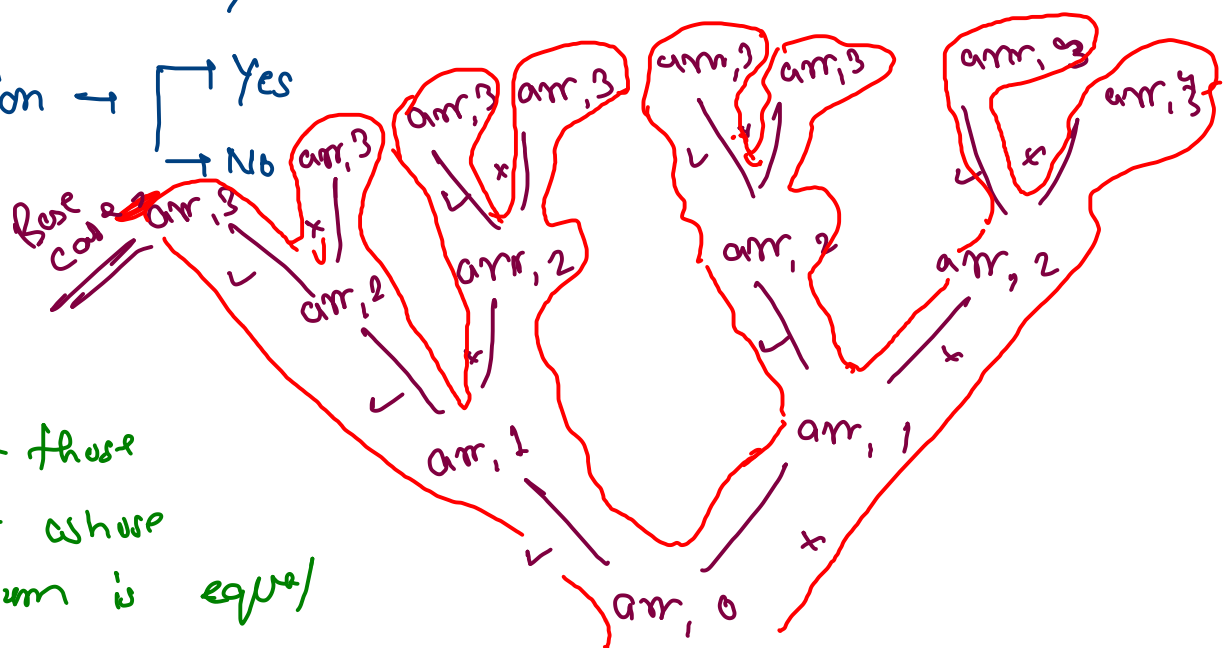
arr → {10, 20, 30}

target = 30

Subseq.

level → Index / character

option → Yes / No



print those subset whose sum is equal to target

index = arr.length.

{10, 20, 30} = 60

{10, 20} = 30

{10, 30} = 40

{10} = 10

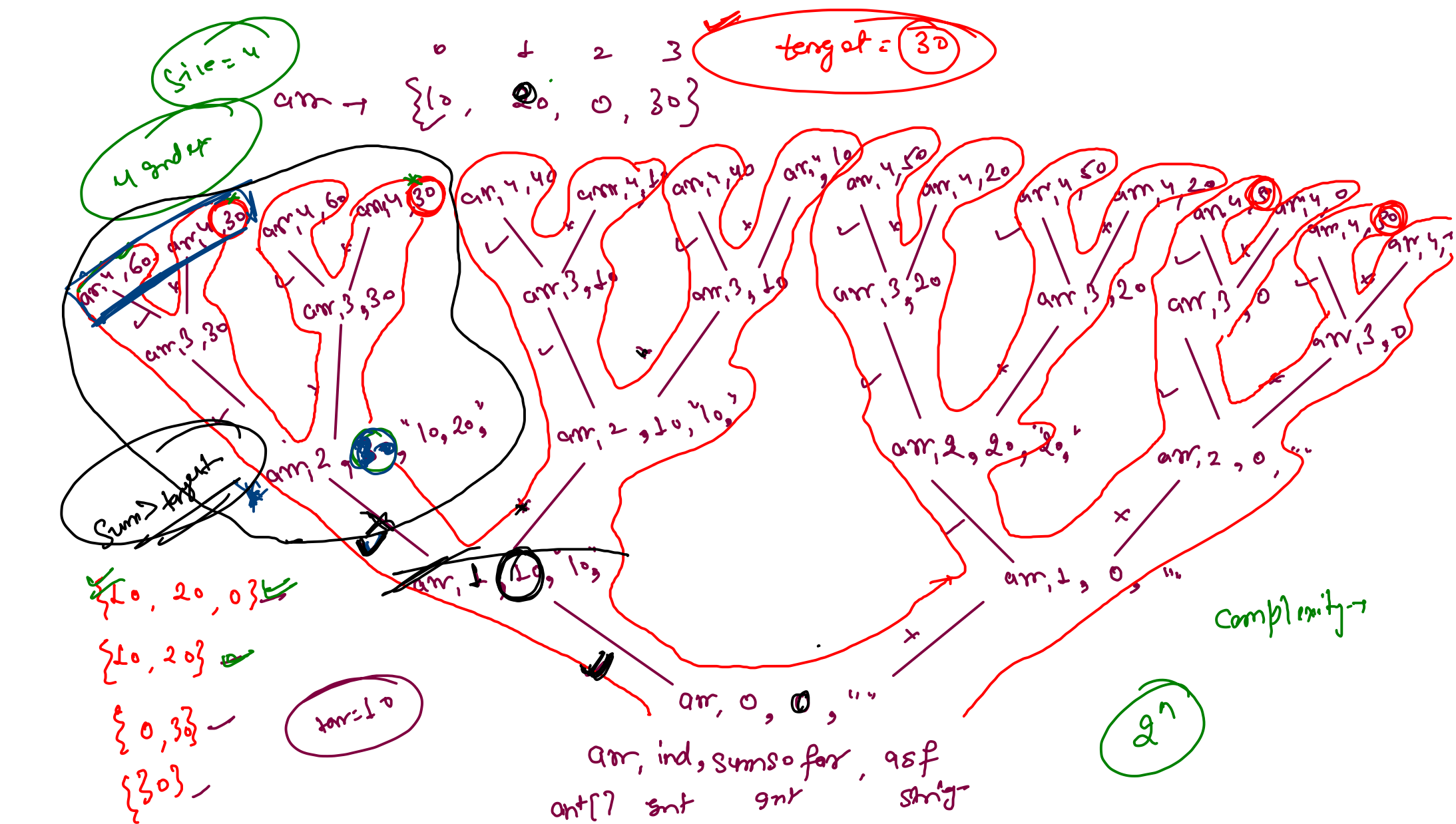
{20, 30} = 50

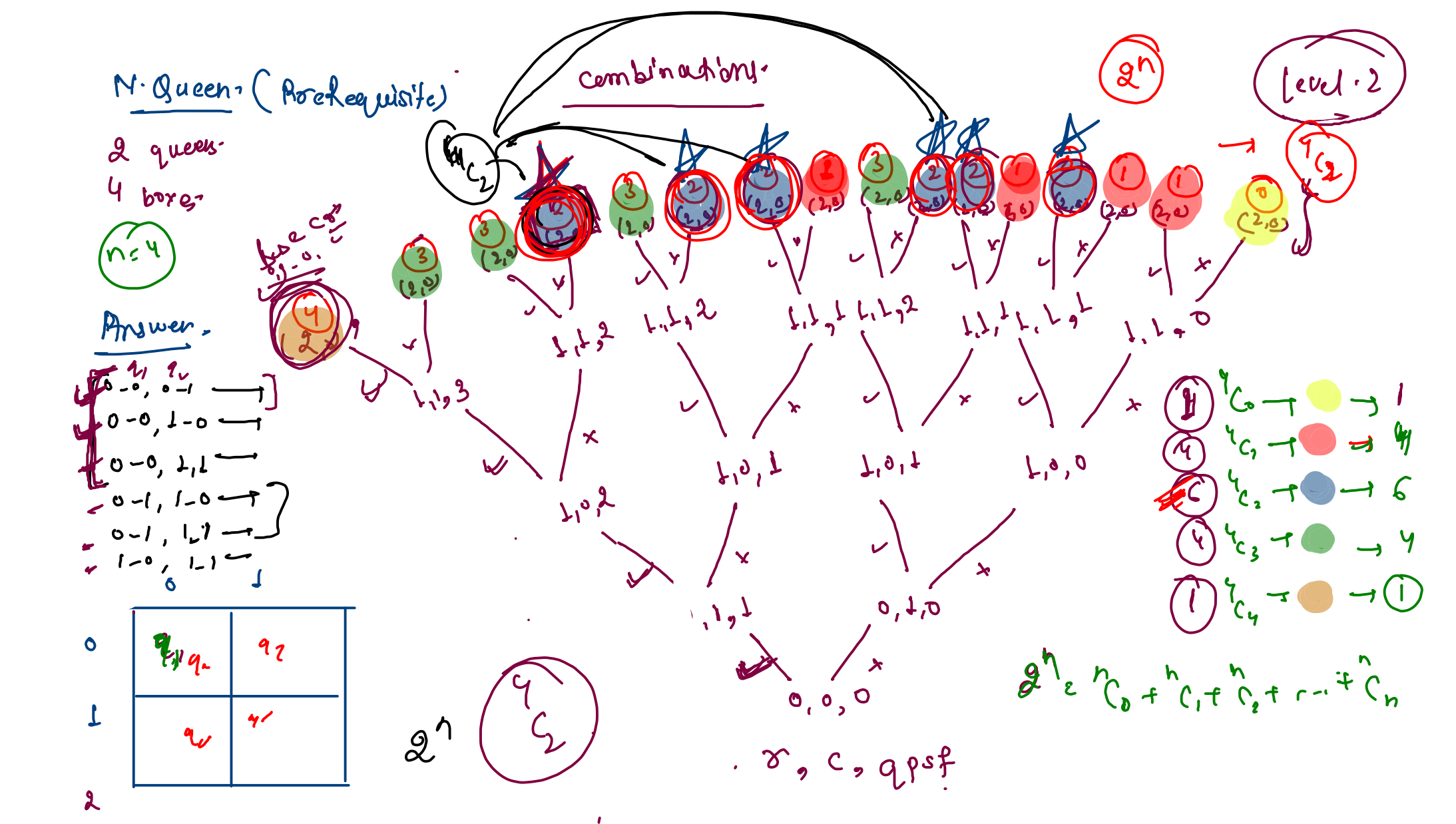
{20} = 20

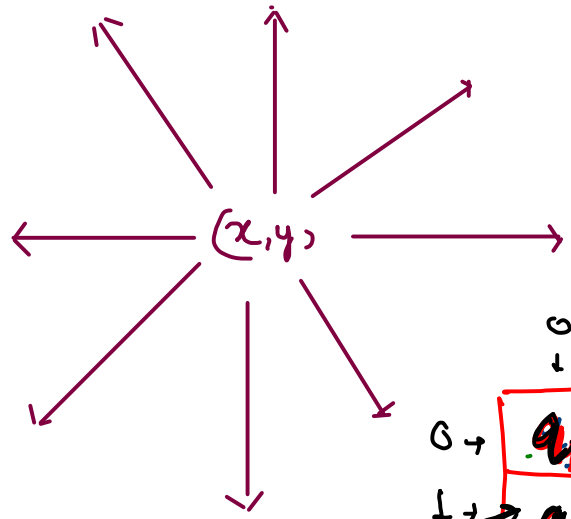
{30} = 30

{} = 0

all subset

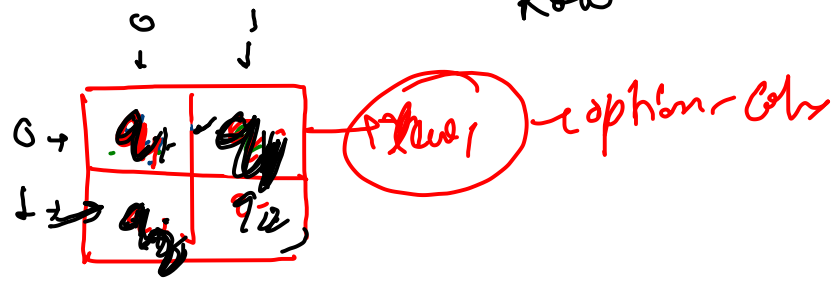

size = 4
4 groups
arr → {10, 20, 0, 30}
0 1 2 3
target = 30
arr,4,60
arr,4,30
arr,4,60
arr,4,30
arr,3,30
arr,3,30
arr,4,40
arr,4,10
arr,4,40
arr,4,10
arr,4,50
arr,4,20
arr,4,50
arr,4,20
arr,4,0
arr,4,0
arr,3,40
arr,3,10
arr,3,20
arr,3,20
arr,3,0
arr,3,0
arr,2,"10,20,"
arr,2,10,"10,"
arr,2,20,"20,"
arr,2,0,"
arr,1,"10,"
arr,1,0,"
arr,0,0,""
Sum > target
{10, 20, 0}
{10, 20}
{0, 30}
{30}
tar=10
arr, ind, sumsofar, asf
ant[] int ant string
Complexity →
2^n

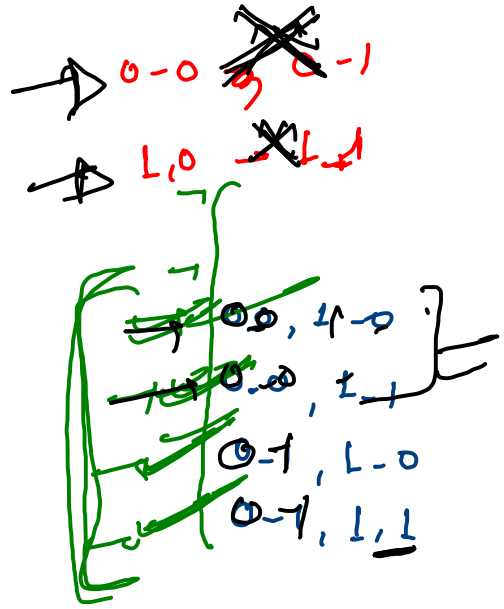

N-Queen (Prerequisite)
2 queens
4 boxes
n=4
Answer
combinations
level-2
2^n
$^4C_2$
$^nC_0 + {}^nC_1 + {}^nC_2 + \dots + {}^nC_n$
2^n $^4C_2$
r, c, qpsf

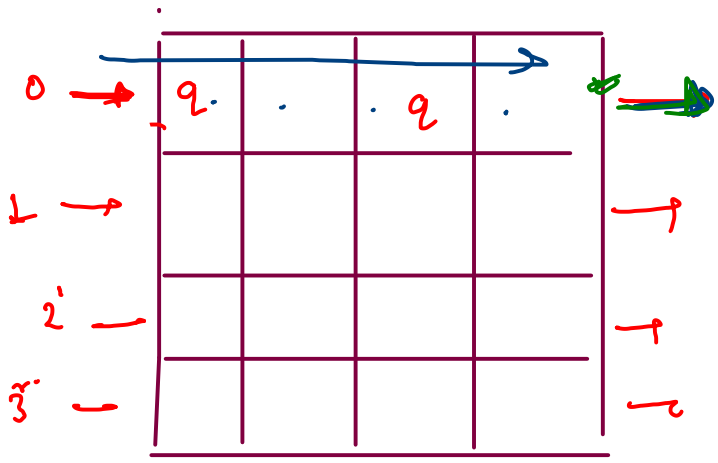


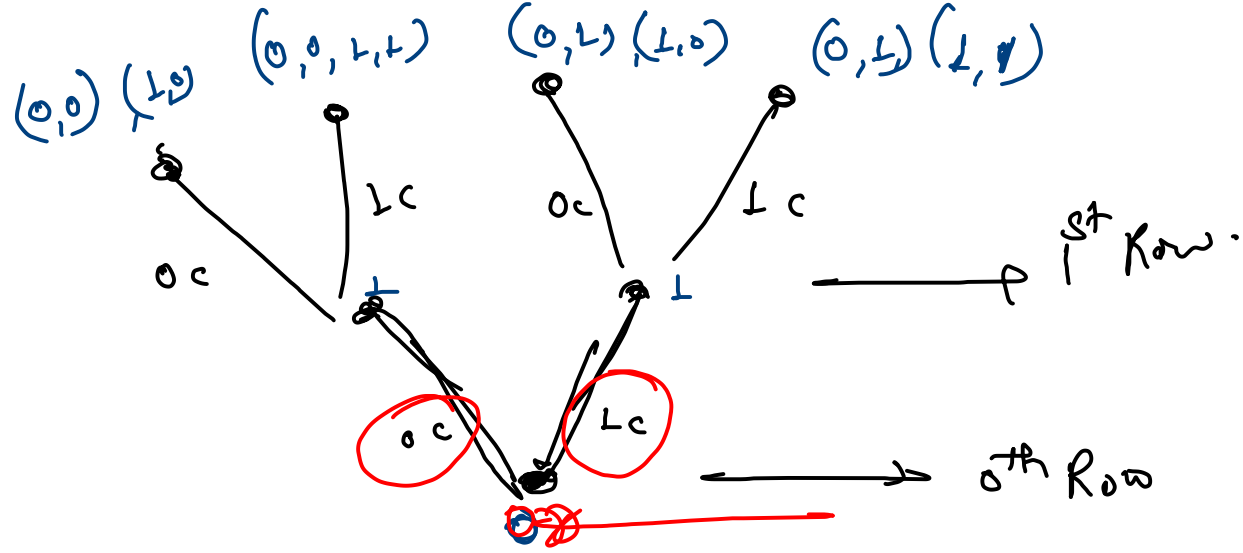
$n \times n$

C_n — { possibility in which 2 queen are placed in same Row }

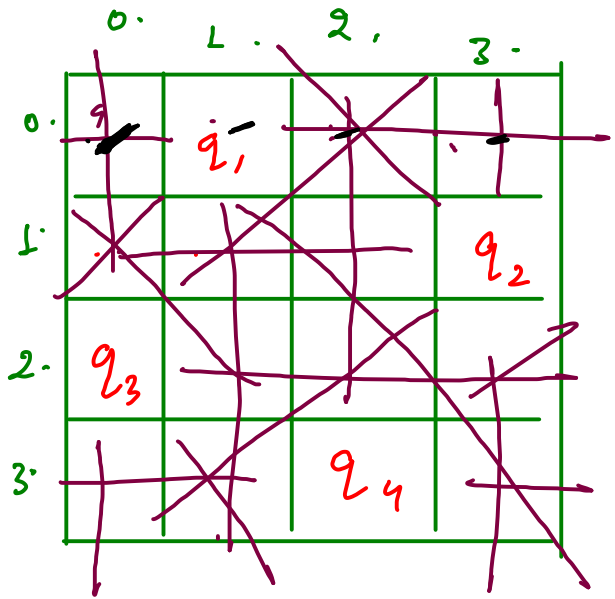




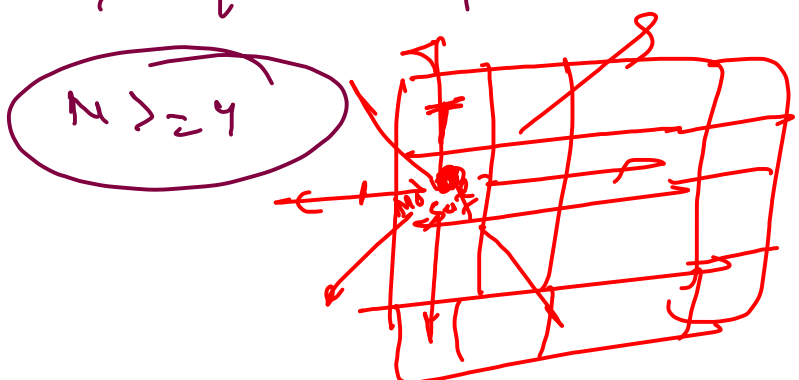




You have N X N chess board print all possibilities to place 'N' queens so that No two queens intersect each other.

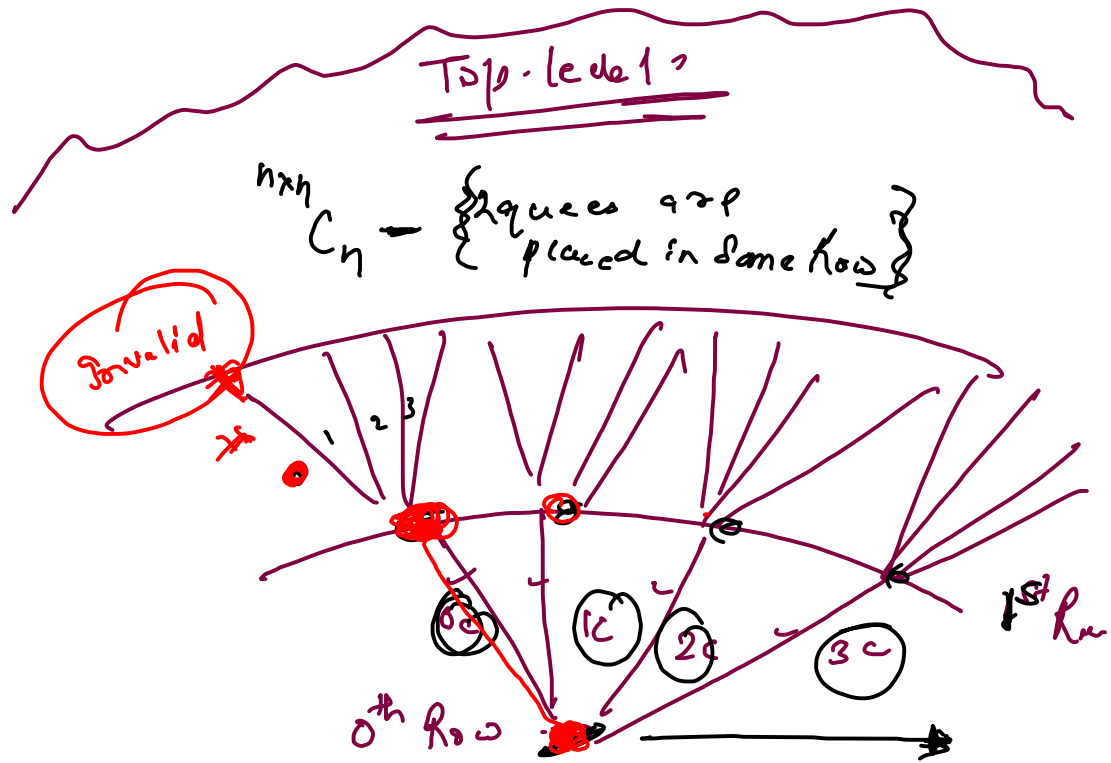


N×N, N queens will place

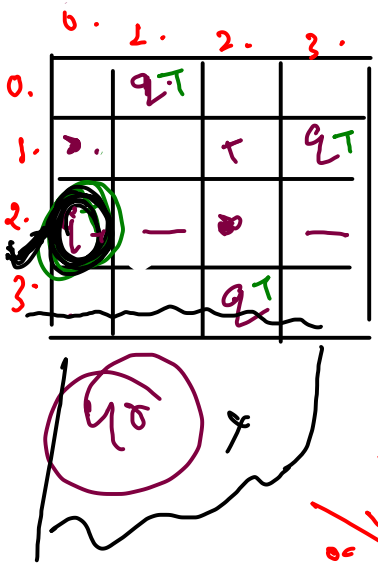


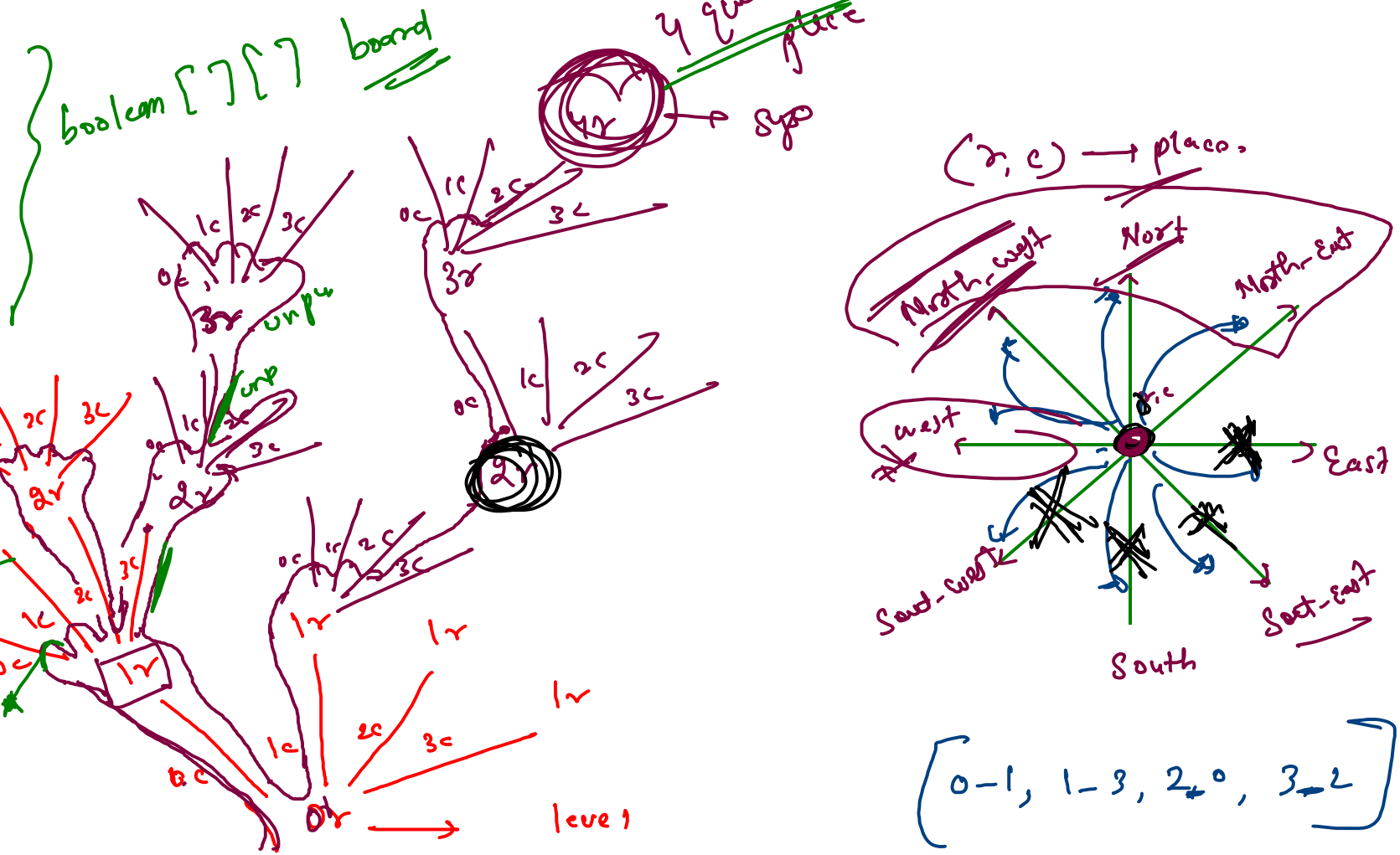
Top-level:

$^{n \times n}C_n$ – {queens are placed in same row}



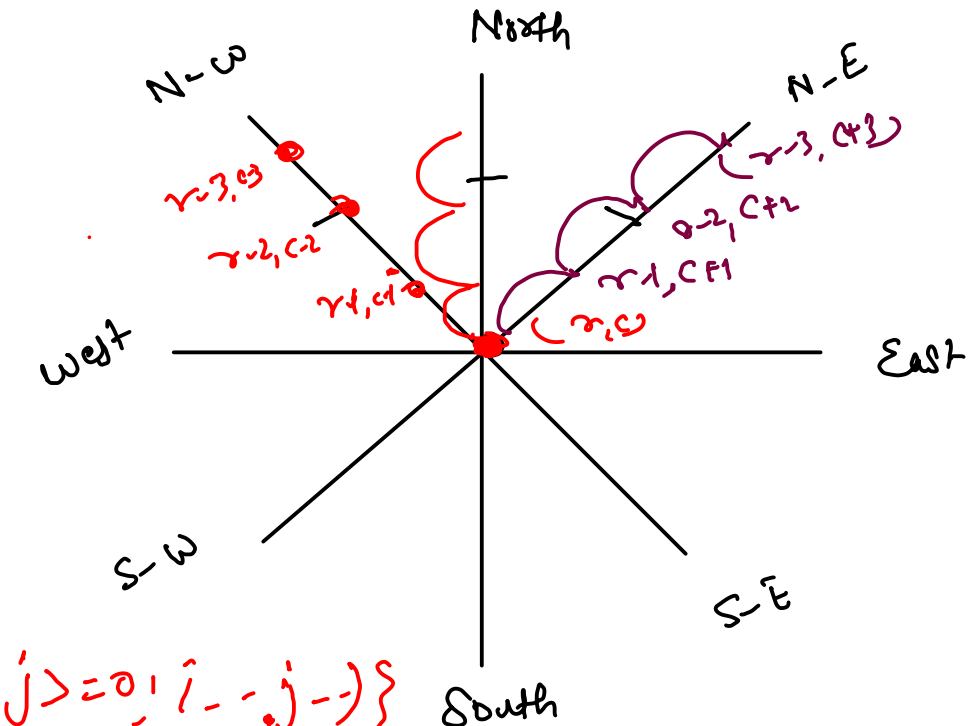
* prevent queens from Intersection.
* prevent queen to place in same Row





Next

```
for(int i = r; i >= 0; i--){
    if (board[i][c] == true){
        return false;
    }
}
```



N-W

```
for(int i = r, j = c; i >= 0 && j >= 0; i--, j--){
    if (board[i][j] == true){
        return false;
    }
}
```

N-E

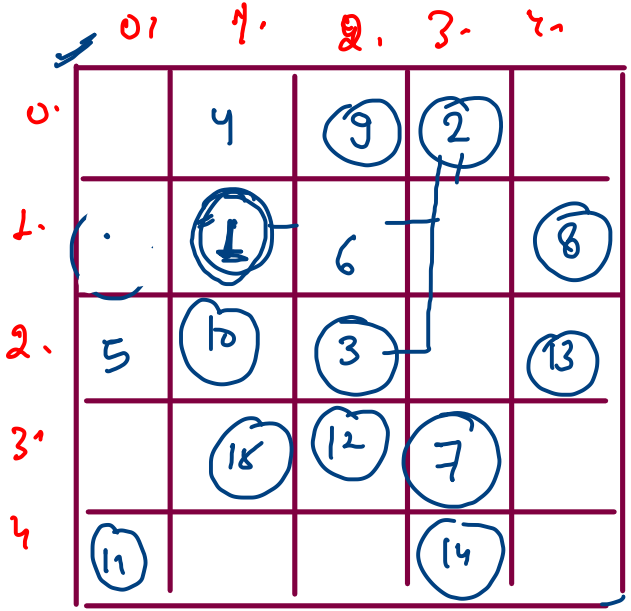
```
for(int i = r, j = c; i >= 0 && j < board[0].length; i--, j++){
    if (board[i][j] == true)
        return false;
}
```

At the end return true;

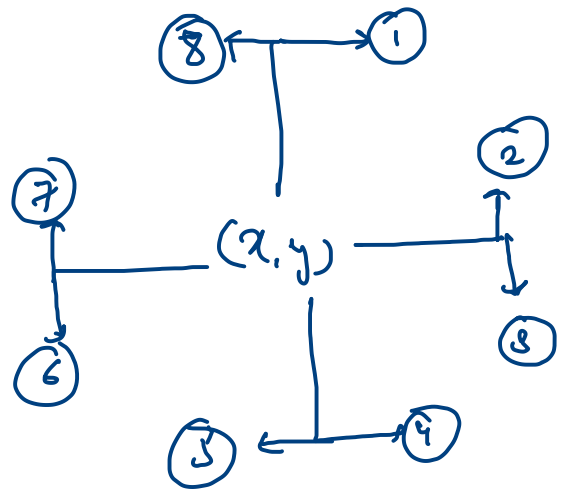
Knights Tour.

Initial coordinate.

Visit all the blocks of board from knight move without visiting any block twice.



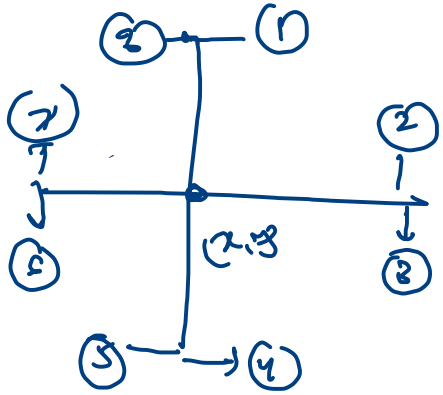
knight —

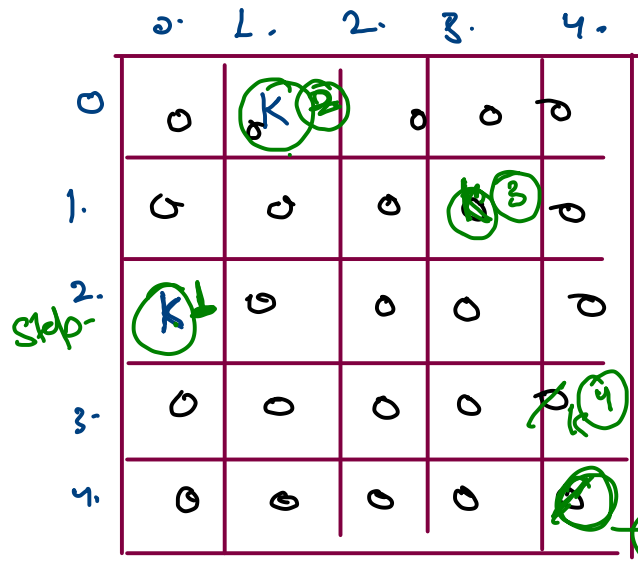


Knights Tour →

n → n×n.

r -] Starting
c -] Coordinates.

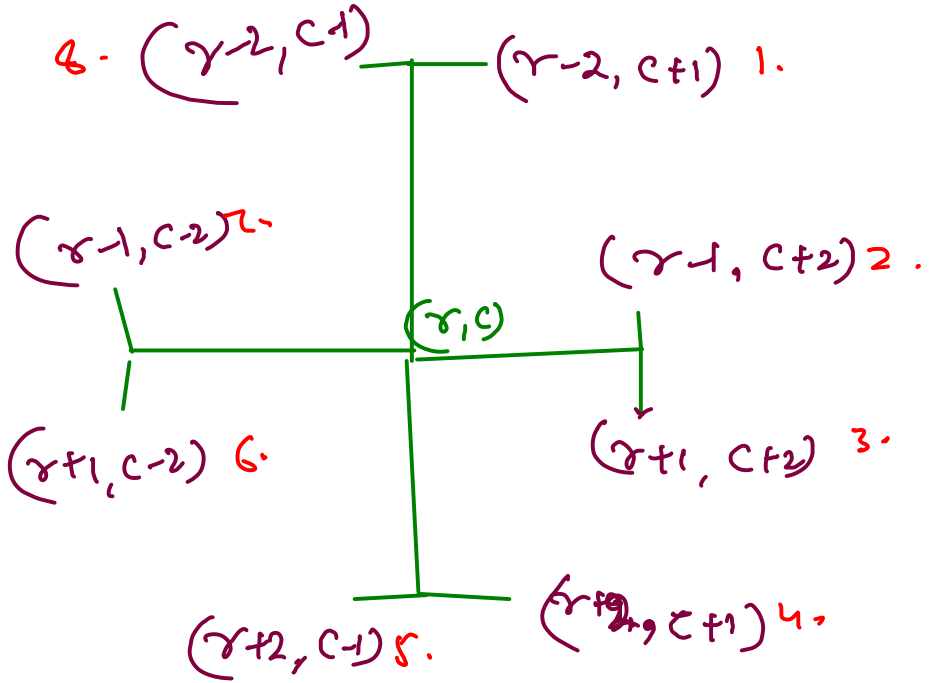




if (board [r] [c] == 0)
unvisited.

step == n * n

25 → Base case



rdir → -2, -1, 1, 2, 2, 1, -1, -2

cdir → +1, 2, 2, 1, -1, -2, -2, -1