R Rooms and Q queens :-

1. Room R, Q queens. Combination.
2. Room R, Q queens. Permutation.

$R = 4$, $Q = 3$ (object)

combination.

①

0	1	2	3
q_1	q_2	q_3	

6 permutation

q_1	q_2		q_3

6 Permutation

q_1		q_2	q_3

6 Permuta

	q_1	q_2	q_3

6 Permutation

$^4C_3$

$\frac{4!}{1!\,3!}$

4

Total Per = 24

②

$^4P_3$

$= \frac{4!}{1!}$

$= 4 \times 3 \times 2 \times 1$

$= 12 \times 2 = 24$

Permutation of single combination

1	q_2	q_3	q_1	
2	q_2	q_1	q_3	
3	q_1	q_2	q_3	
4	q_1	q_3	q_2	
5	q_3	q_1	q_2	
6	q_3	q_2	q_1	

Permutation of R Rooms and Q queens:

(In permutation queen number is also important in placing)

$R = 3 \quad Q = 2.$

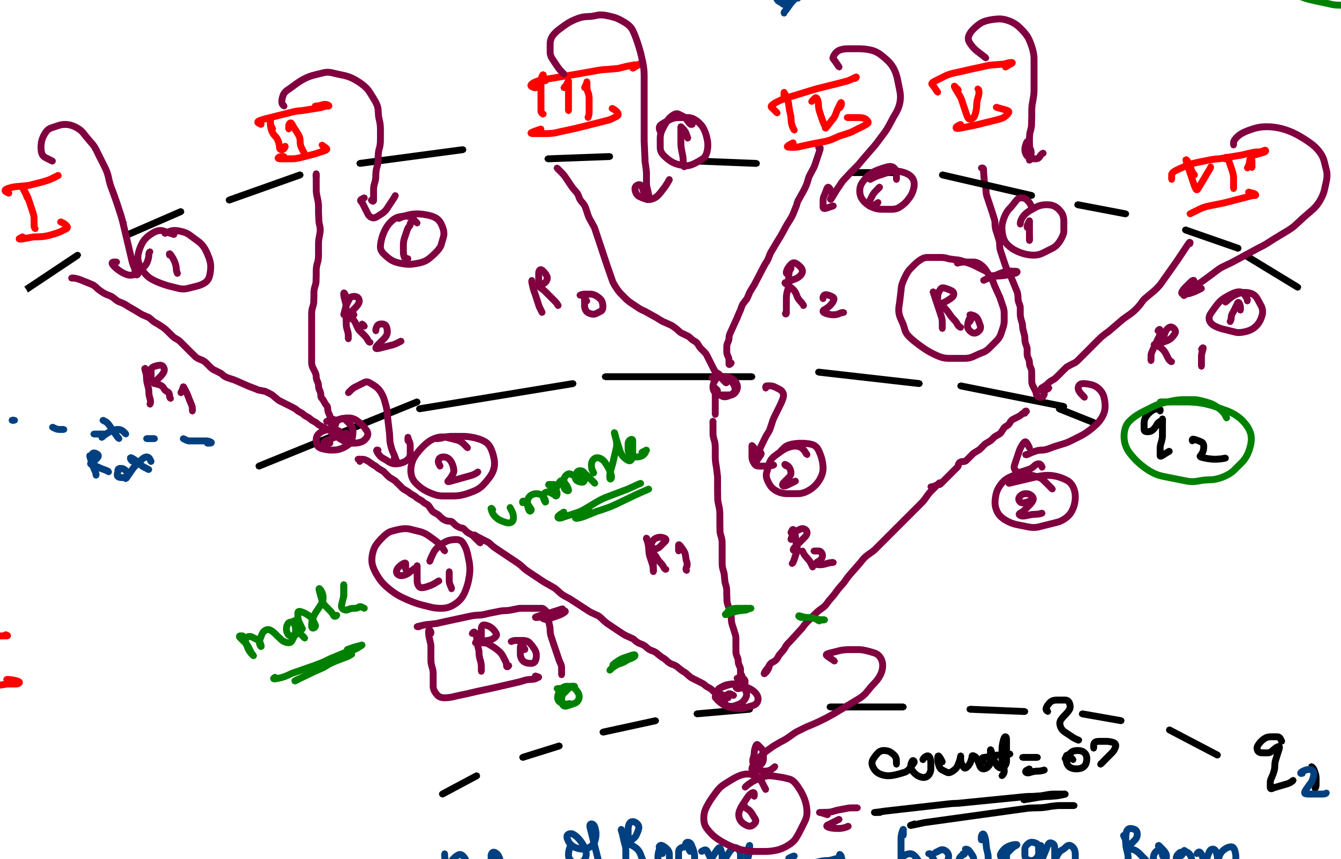
$^{3}P_{2} = \frac{3!}{1!} = 3 \times 2 = 6$

Level - option.

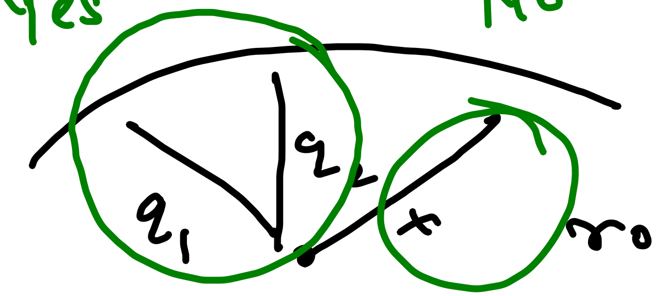
Queens. Rooms.

→ Room - queen. →

R_0	R_1	R_2	
q_1	q_2	—	I
q_1	—	q_2	II
q_2	q_1	—	III
q_2	—	q_1	V
	q_1	q_2	IV
	q_2	q_1	VI



subseq



→ subseq

which queen is placed → combination

queen ←

qpsf = 0 → queen

no queen

qpsf = queen placed so far.

6 = no. of Rooms → boolean Room

no. of queens:

qpsf = 2

print → asf

$R_0 q_1$, $R_1 q_2$ ..

variables.   Initially.

① queen placed so far (qpsf) = 0
② Total queens = q = 3.
③ Rooms boolean size - (R=4) size.
④ answer so far (asf) append,

q=2
R=3

$$^3C_2 = \frac{3!}{2!\,1!} = \frac{3\times 2!}{2!} = 3$$

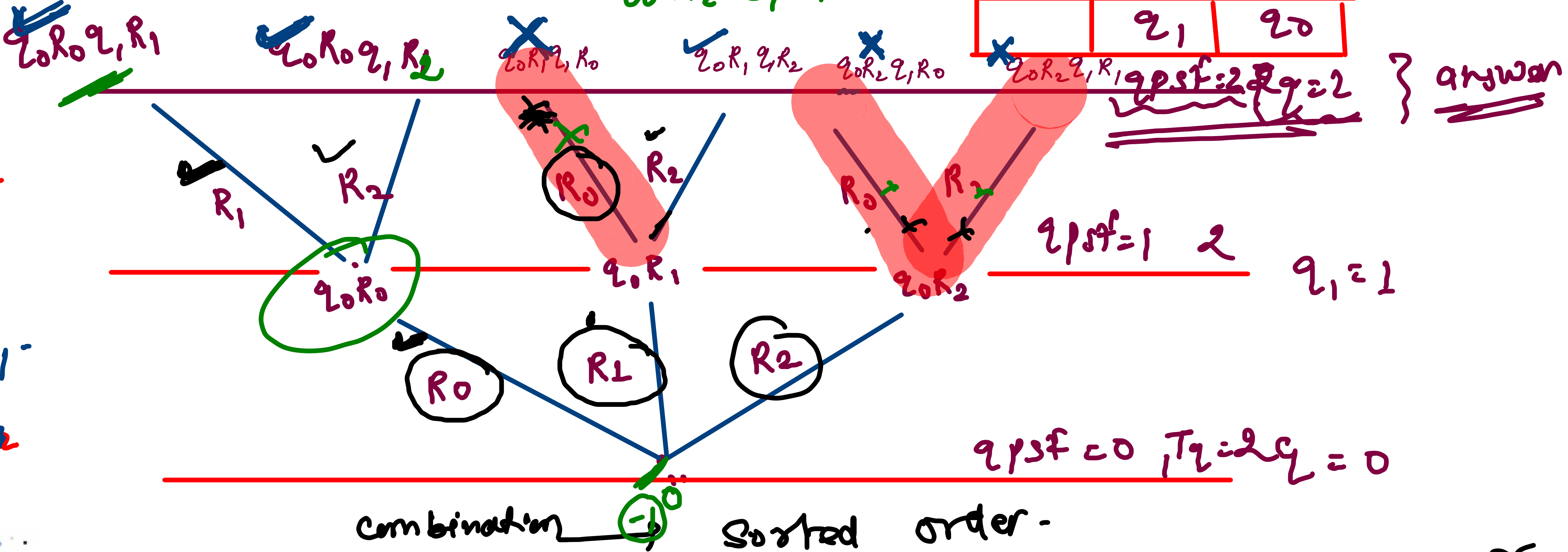
q=2
R=3

Combination,

R0	R1	R2
q0	q1	-
q0	-	q1
-	q0	q1

q0R0 q1R1
q0R0 q1R2
q0R1 q1R0
q0R1 q1R2
q0R2 q1R0
q0R2 q1R1

R0	R1	R2
q0	q1	
q0		q1
q1	q0	
	q0	q2
q1		q0
	q1	q0



output.txt

```
R0Q0, R1Q1, .
R0Q0, R2Q1, .
R1Q0, R2Q1, .
```

last room occupied = r.

For current level - try from (r+1)

N Queens → $n \times n$ → chessboard

n - queens placed.

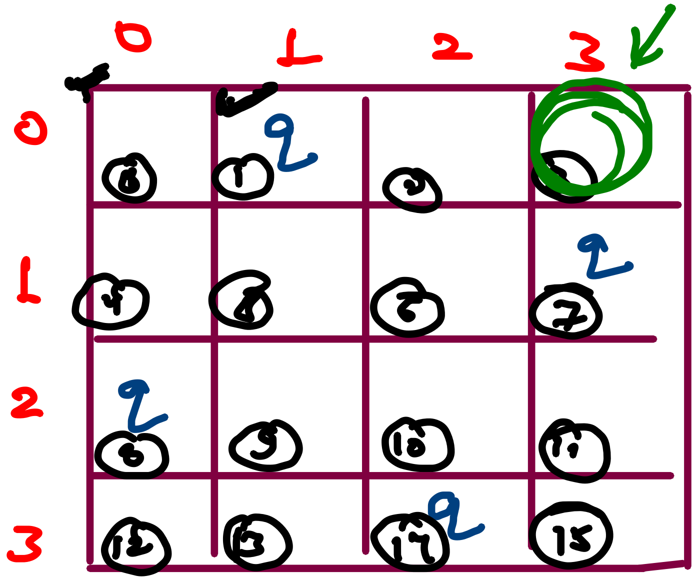
$n \geq 4$.

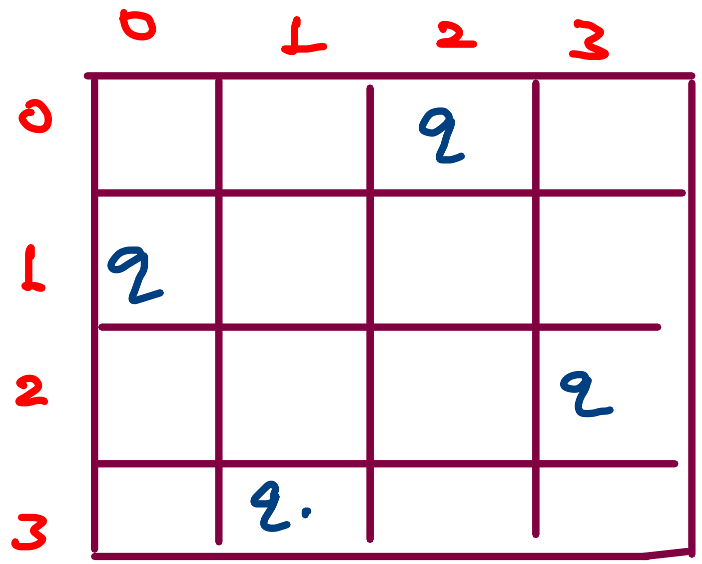
level - queen.

option - box

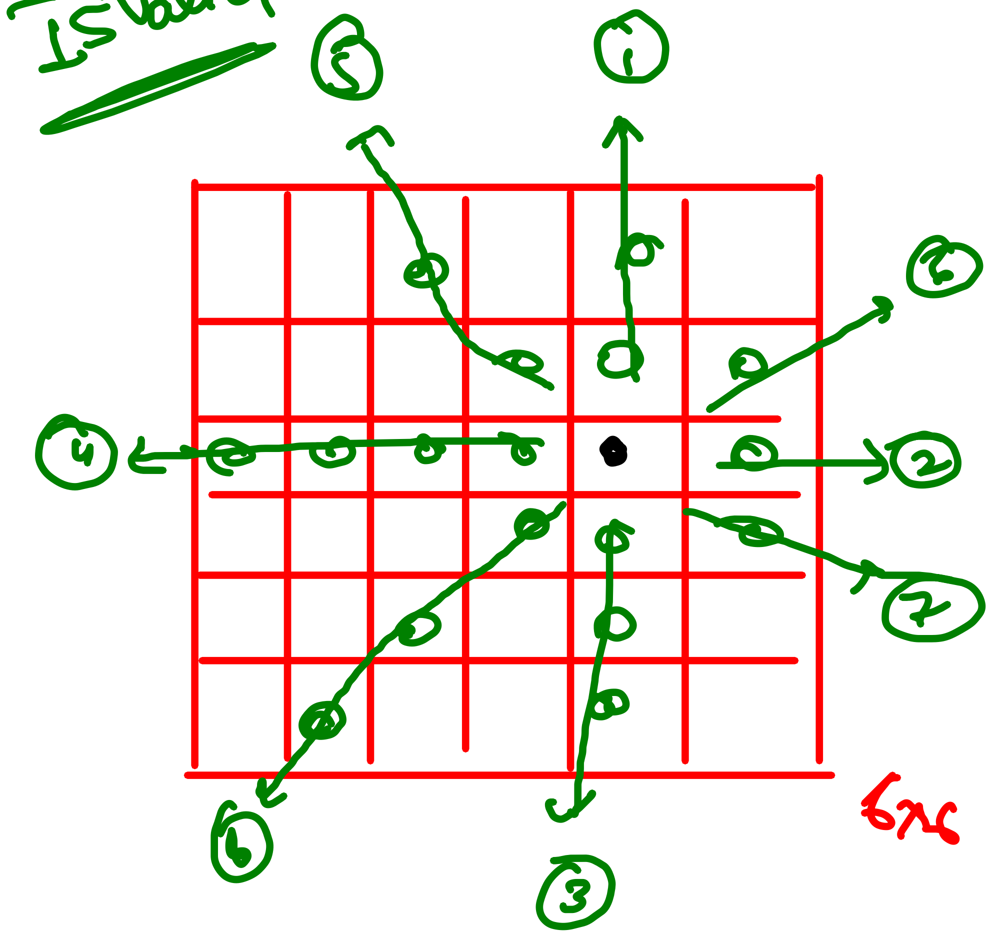
combination.

→ last box





IsValid



Convert 2D array into 1D →

assume: all the boxes are 1D

box no.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Actual →	(0,0)	(0,1)	(0,2)	(0,3)	(1,0)	(1,1)	(1,2)	(1,3)	(2,0)	(2,1)	(2,2)	(2,3)	(3,0)	(3,1)	(3,2)	(3,3)

16 box

4 queen

n=4

n=4

can we extract Row & Col from box no?

11 box no. →

R = 2

C = 3

Relation b/w box No. & (Row & Column)

R = box no / Total Column.

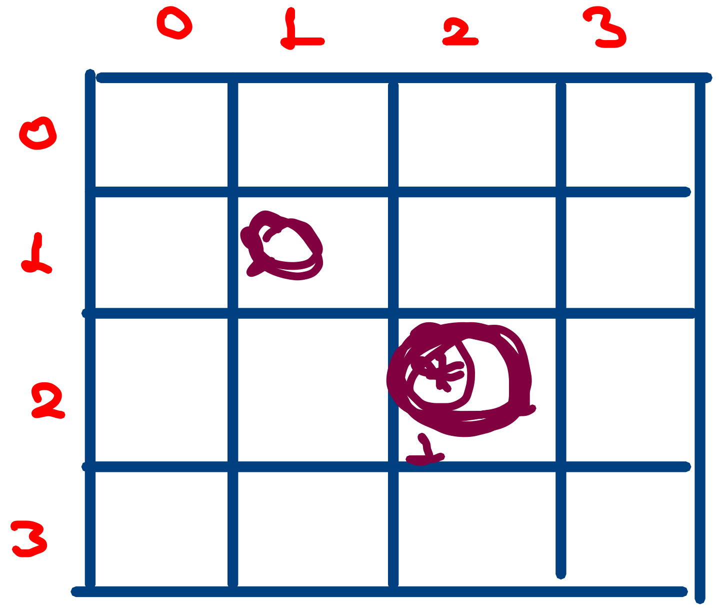
C = box no % Total Column

Knights Tour → Knight can visit all the cells exactly once.

r, c
↓ ↓
(2 0)

level → Base case

option → calls



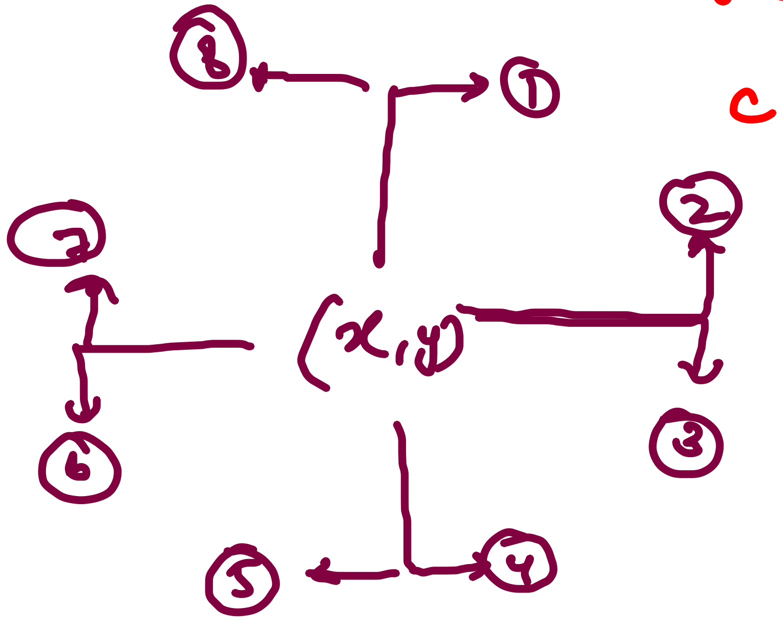
jsf

	0	1	2	3	4
0	25	2	13	8	23
1	12	7	24	3	14
2	1	18	15	22	9
3	6	11	20	17	4
4	19	16	5	10	21

$r \times r = 25$

r = 2

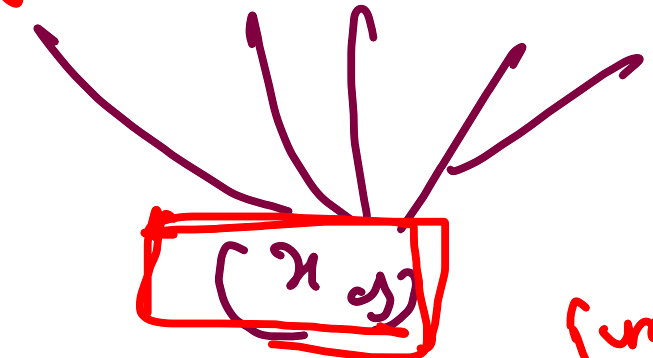
c = 0



Signature → option → jumps & angles → also Responsible for calls

unvisited && valid

out of range



level → jsf, jump so far

jsf == n×n

PRINT

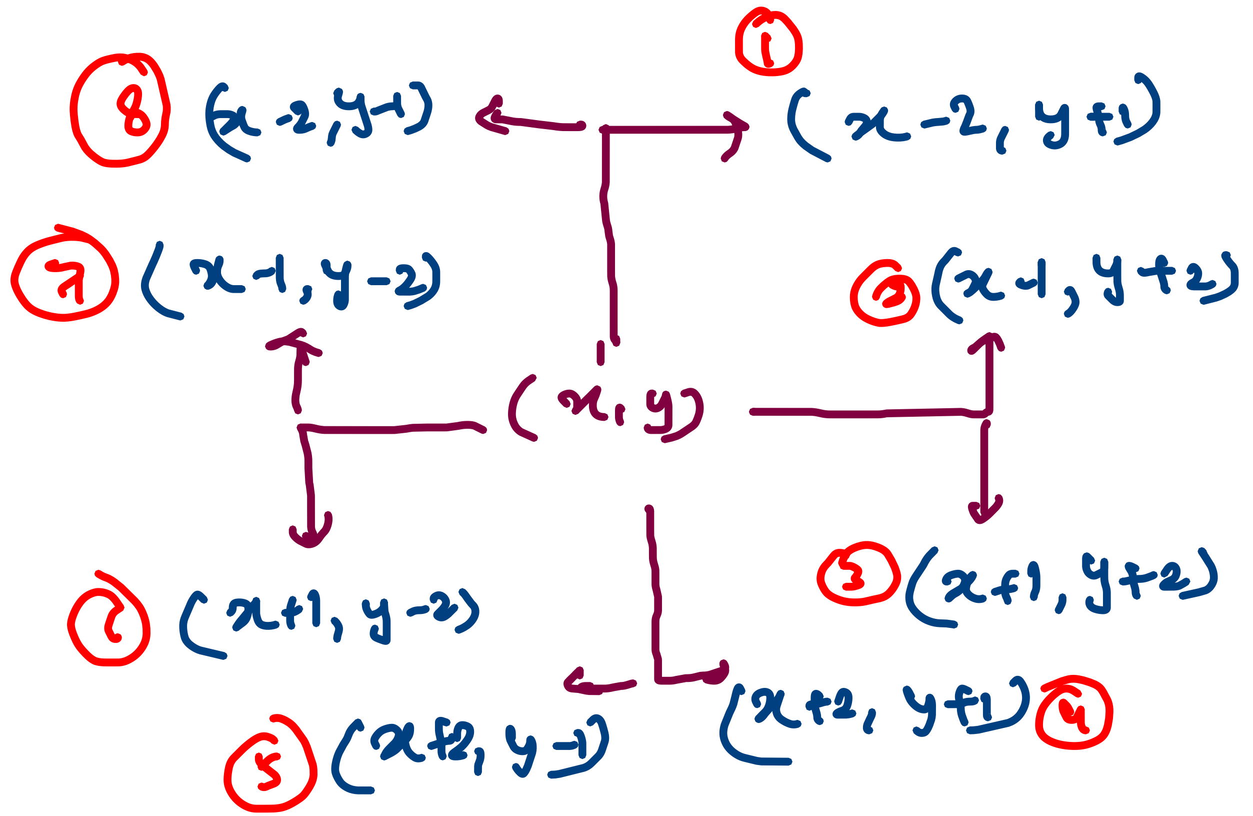
level → cell visited so far.

```
if(cell visited so far == n*n) {
    Base case; printboard;
}
```

options → Eight calls.

① Position valid.
② unvisited

xdir = {-2, -1, 1, 2, 2, 1, -1, -2}
ydir = {1, 2, 2, 1, -1, -2, -2, -1}



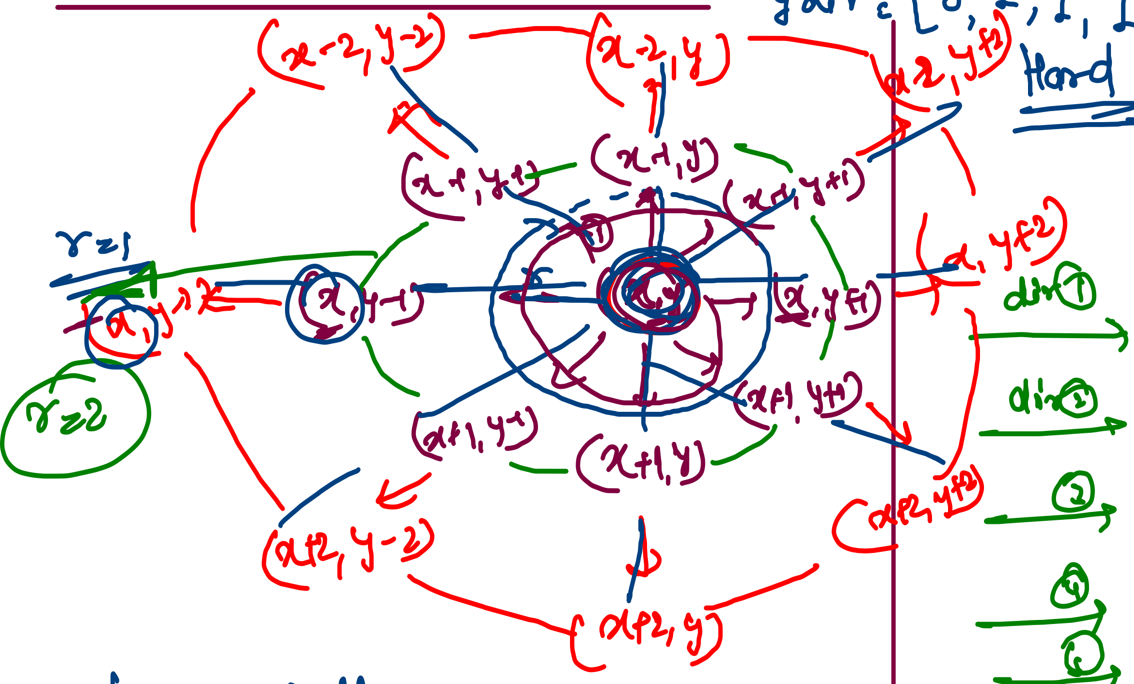
How to think about isValid

xdir = [-1, -1, 0, 1, 1, 1, 0, -1]
ydir = [0, 1, 1, 1, 0, -1, -1, -1]

```java
public static int[] xdir = {-1, -1, 0, 1, 1, 1, 0, -1};
public static int[] ydir = {0, 1, 1, 1, 0, -1, -1, -1};

public static boolean isValid(boolean[][] board, int x, int y) {
    for(int r = 1; r < board.length; r++) {
        for(int d = 1; d <= 8; d++) {
            int rr = x + (xdir[d] * r
            int cc = y + (ydir[d] * r

        }
    }
    return false;
}
```

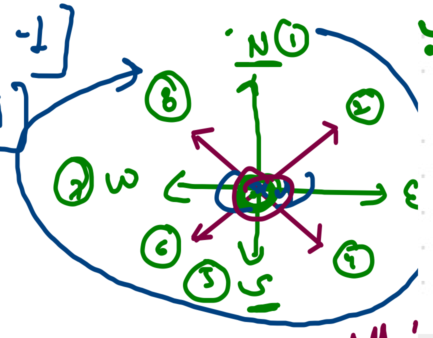


Hard

(x, y)

dir 1
dir 2
3
4
5
6
7
8

loop



Mistake - probably

$16_{C_4} = 1820$

16_{P_4}

loop → radius

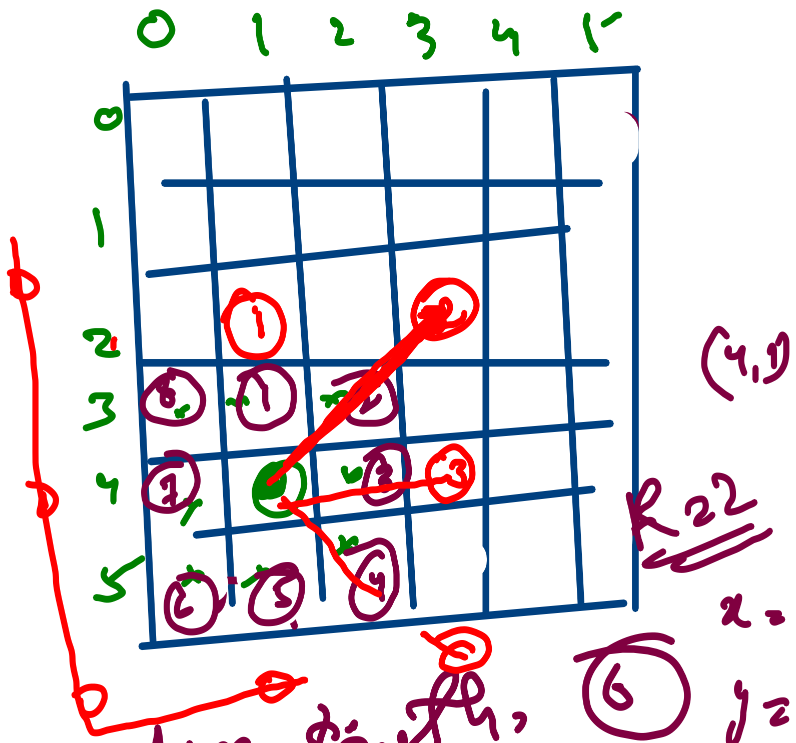
radius = 1

max radius = column size

R=1 4,1

	1	2	3	4	5	6	7	8
x →	3	3	4	5	5	5	4	3
y →	1	2	2	2	1	0	0	0

$\frac{16!}{12!}$ = 16x15x14x13



column.length

R=2

(4,1)

x = 2, 2, 4, 6, 6, 6, 4, 2
y = 1, 3, 3, 3, 1, -1, -1, -1

beyond the range

beyond the range

R=2

CRYPTO ARITHMATIC →

SEND
+ MORE
MONEY

9 5 6 7
+ 1 0 8 5
1 0 6 5 2

string1 → SEND
string2 → MORE
string3 → MONEY

valid string

S → 9
E → 5
N → 6
D → 7
M → 1
O → 0
R → 8
Y → 2

generate all possible mapping.

Map character with digits (0 to 9)

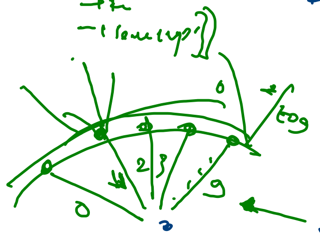
unique Mapping. S → 1

M →

SENDMORY
characters Map } 0 to 9.

time

D.S
→ fn
→ (lower cap?)



S
M
D
N 0 to 9
E
S

boolean.
10 size

26 size

a b c - - - z

0

A B C D E F G H
3 0 2 1 6

D F G
→ $\boxed{2\ 1\ 6}$

DFG → 216

2*10 + 1 = 21*10 + 6 = 216 return

num = 0
number*10 + map[i]

num = num*10 +

CRYPTOARITHMATIC →

SEND
\+ MORE
MONEY

E → 5
N → 6
D → 7
M → 1
S → 9
O → 0
R → 8
Y → 2

9 5 6 7
\+ 1 0 8 5
1 0 6 5 2

string 1 → SEND
string 2 → MORE
string 3 → MONEY

valid string

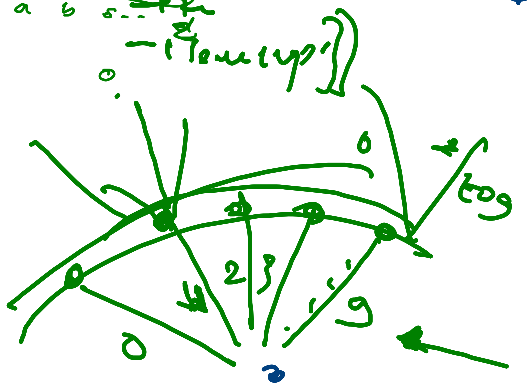
generate all possible mapping
SENDMORY
Map character with digits (0 to 9)
unique Mapping.
S → 1

M →

D S
time

boolean.
10 size

D
M
D
N 0 to 9
E
S



26 size
a b c . .

D	F	G
2	1	6

A B C D E F G H
3 0 2 1 6

DFG → 216

SENDMORY
character → Map
0 to 9

$2*10+1 = 21*10+6 = 216$
return

num = 0
number*10 + map[i]

num = num*10+

Principle of Mathematical Induction → (PMI) →

$$1+2+3+4+ \dots + n = \frac{n(n+1)}{2}$$ } → Expectations

suppose, it is true for k.

$$1+2+3+ \dots + k = \frac{k(k+1)}{2}.$$ } Faith Small problem will work.

Now check for $k+1$ (Merging of faith & expectation in R.H.S.)

L.H.S. $\underbrace{1+2+3+ \dots + k}+ k+1$

$$= \frac{k(k+1)}{2} + (k+1) = (k+1)\left[\frac{k}{2}+1\right]$$

$$= \frac{(k+1)(k+1+1)}{2}$$

L.H.S. $$= \frac{(k+1)((k+1)+1)}{2}$$

$$= \frac{n(n+1)}{2}.$$

$$= \frac{(k+1)((k+1)+1)}{2}$$

L.H.S = R.H.S.

True.