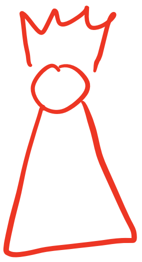
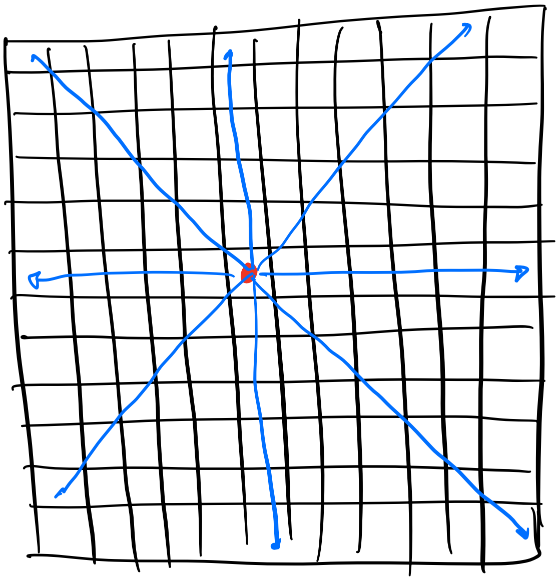
Q

N-Queens

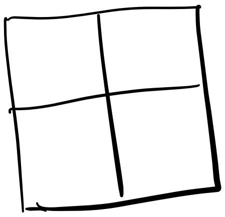
N × N chess board.

Tell the no. of ways of putting N queens on it such that no two queens attack each other!



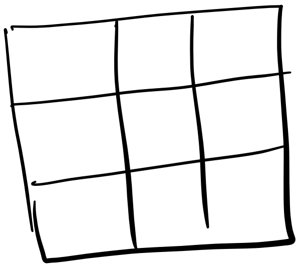


N = 2



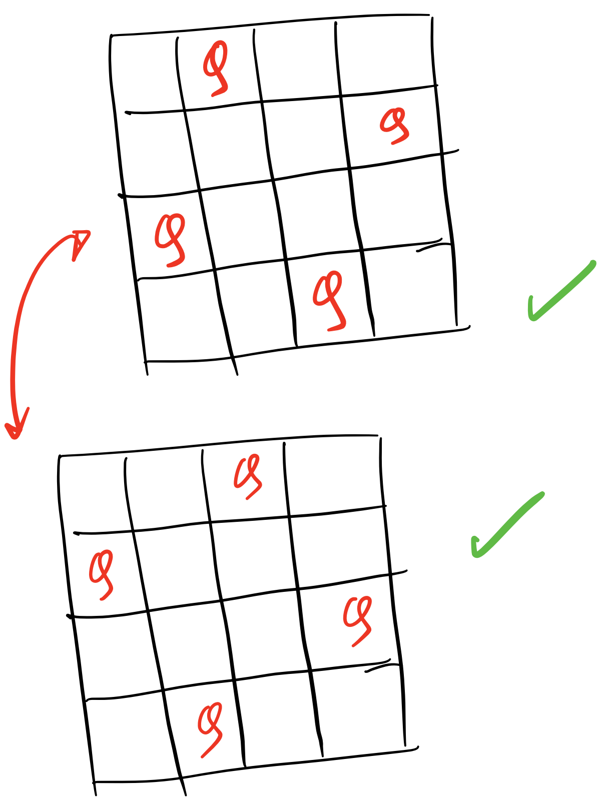
→ 0

N = 3



→ 0

N: 4



	q		
			q
q			
		q	

✓

		q	
q			
			q
	q		

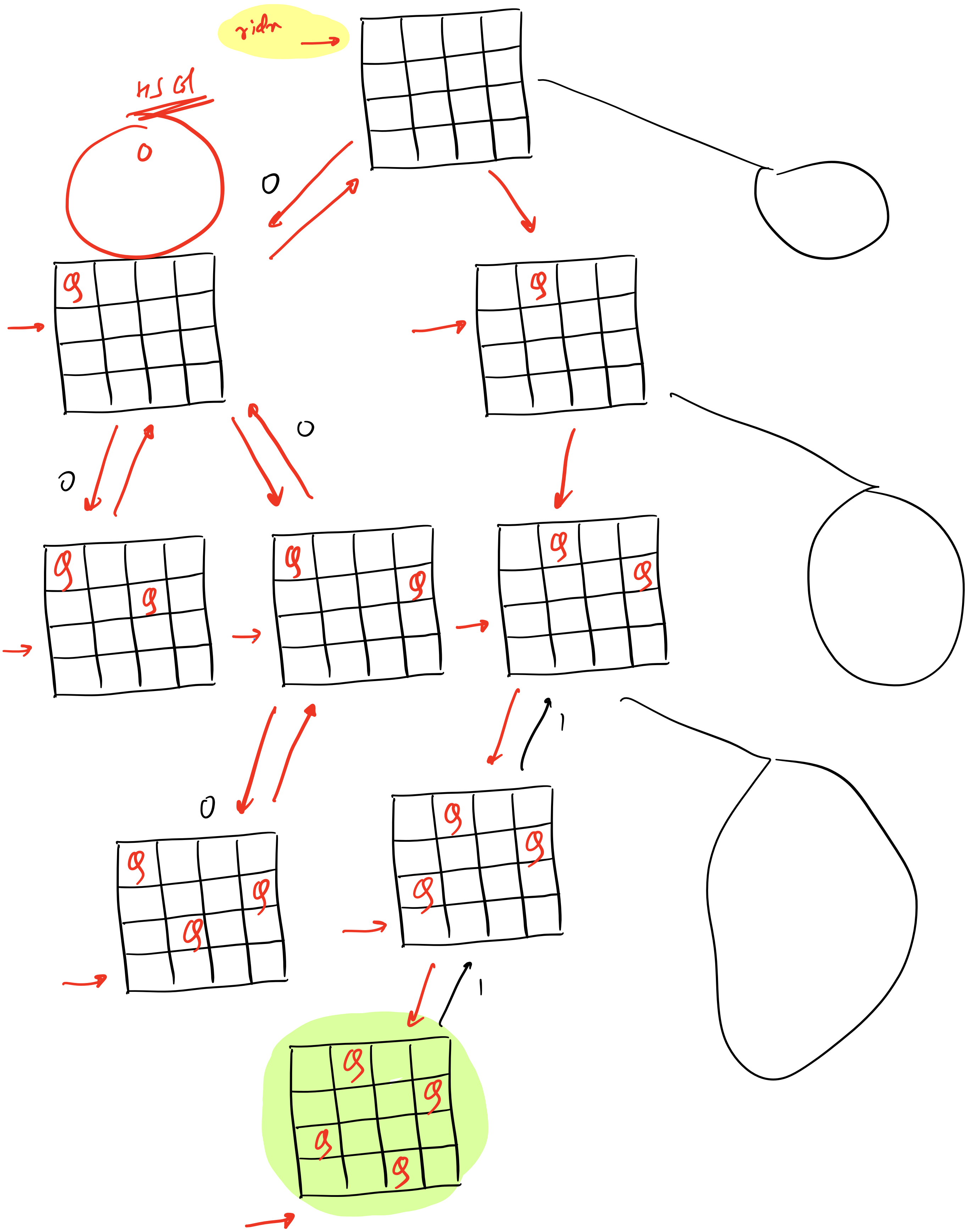
✓

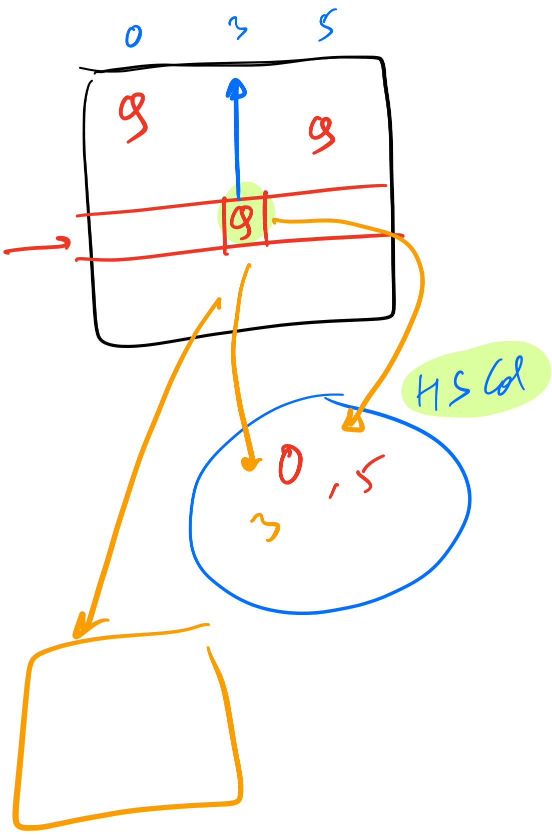
① Obs →

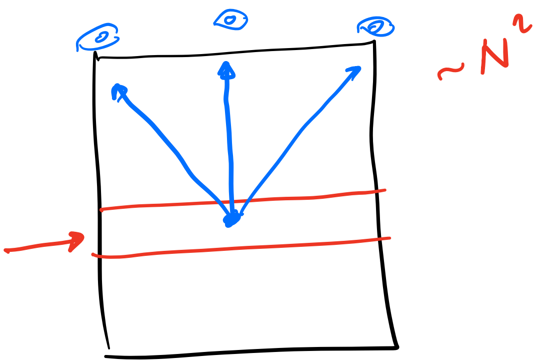
→ No 2 Queens in same Row.

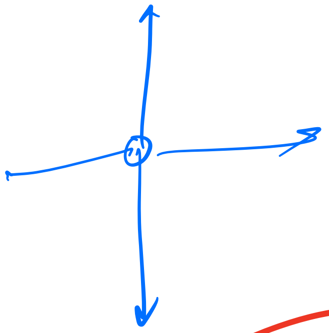
→ ______ Col

→ ______ Diag.

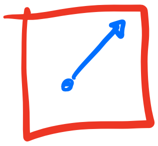

ridx
0
0
0
0
0
1
1
q



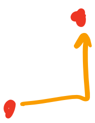




Right Up

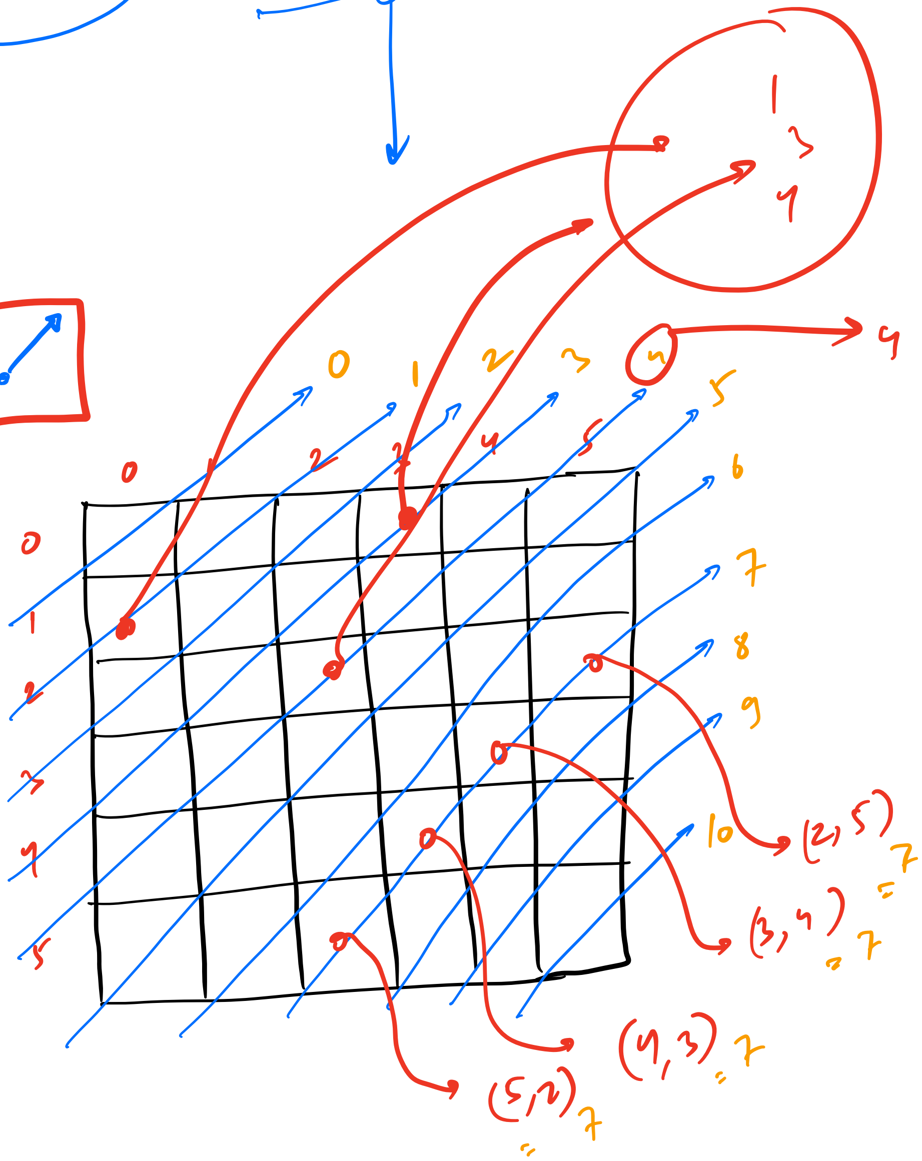


$(x-1, y+1)$ ↘ $x+y$



$(x, y) = x+y$

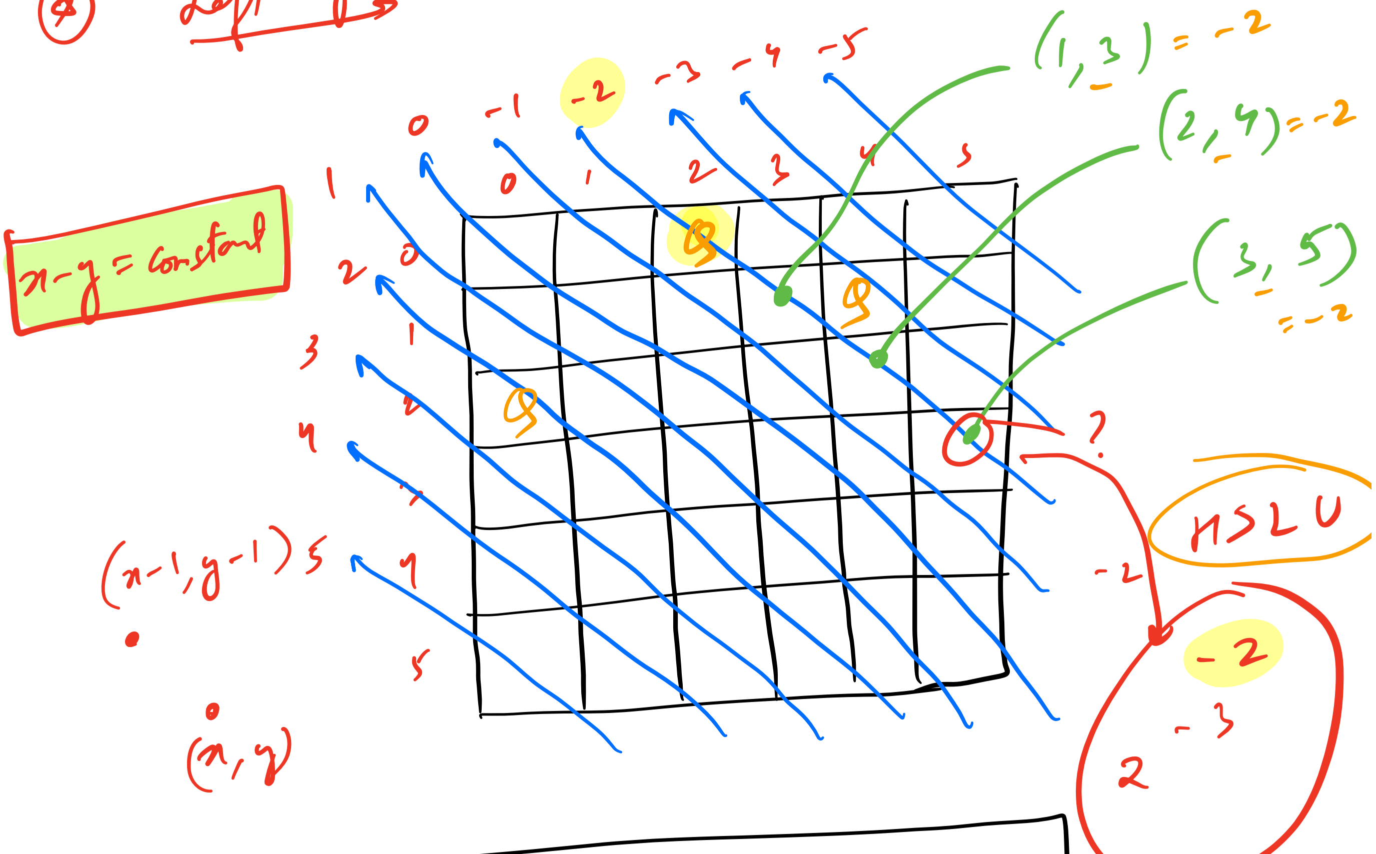
$\boxed{x+y = \text{Constant}}$

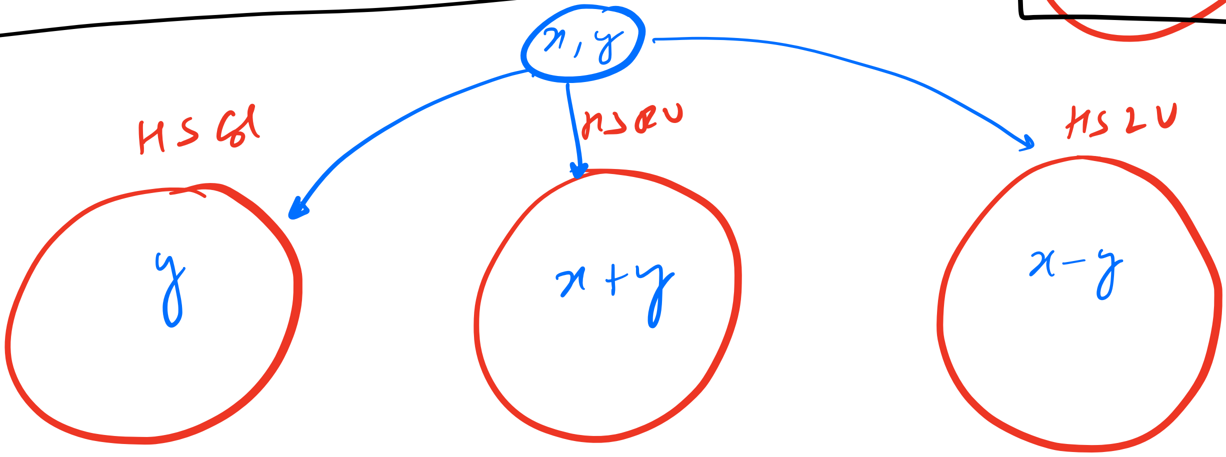


HSRU

~~$x+y$~~

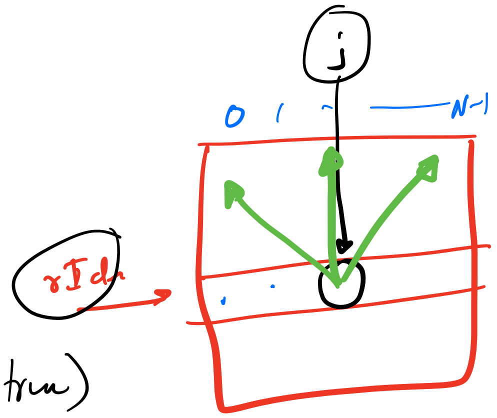
④ Left Up

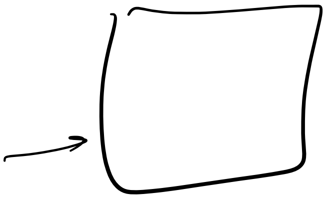




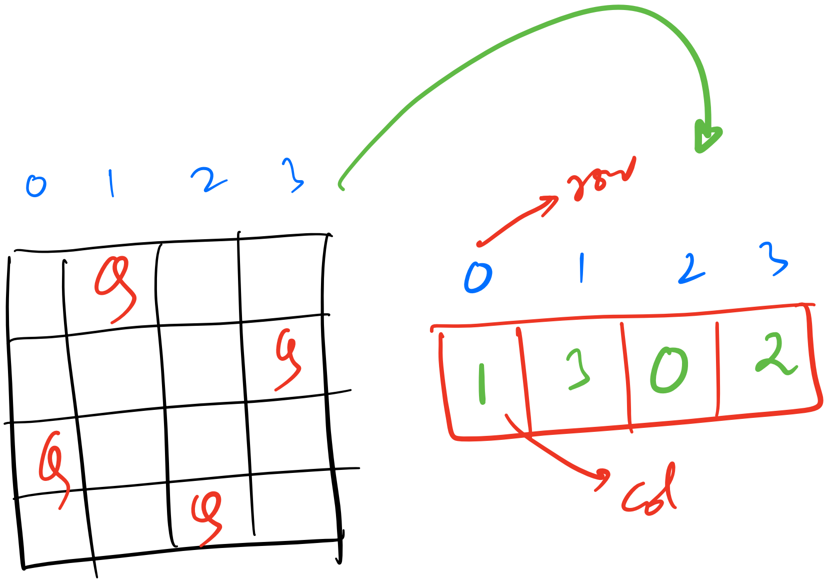
CODE

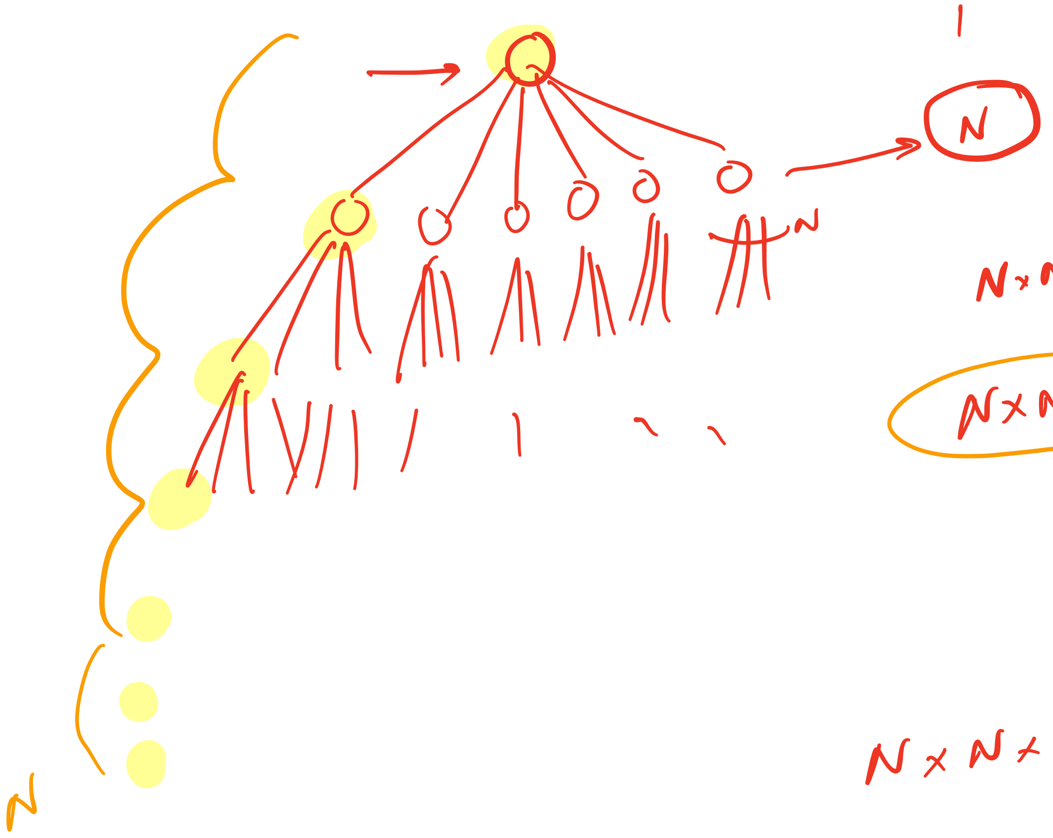
```
char cur[N][N] = {'.'};
Hashset <int> HSCol, HSRU, HSLU;

void gen (rIdx, cur, HSCol, HSRU, HSLU) {
   if ( rIdx == N) {
        print(cur);
         cnt++;
        ret;

   }
   f( j=0; j<N; j++) {
        if ( HSCol. contains(j) == true)
               continue;

        if ( HSRU. contains( rIdx + j) == true)
                continue;

        if ( HSLU. contains ( rIdx-j)== true)
                continue;
do      cur[rIdx][j] = 'Q';
do      HSCol. insert (j);
do      HSRU. insert( rIdx +j);
do      HSLU. insert ( rIdx - j);
        gen( rIdx +1, cur, HSCol, HSRU, HSLU);
```





```
undo → HSLU.remove(rIdx-j);
undo → HSRU.remove(rIdx+j);
undo → HSCol.remove(j);
undo → arr[rIdx][j] = '.';

}

}
```





N×N

N×N×N

N!

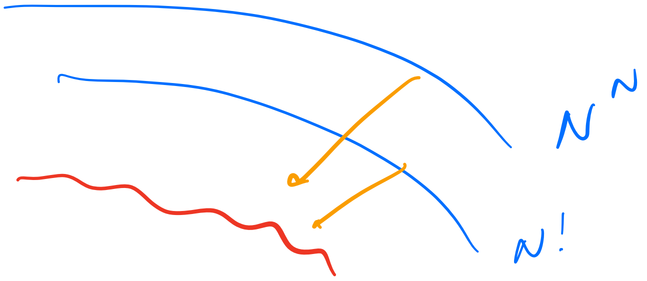
$N \times N \times \text{—} N = N^N$

$\boxed{TC = O(N \cdot N)} \rightarrow N!$

$\times N$

$N <= 10$

15



$\boxed{SC = O(N)}$

Cur

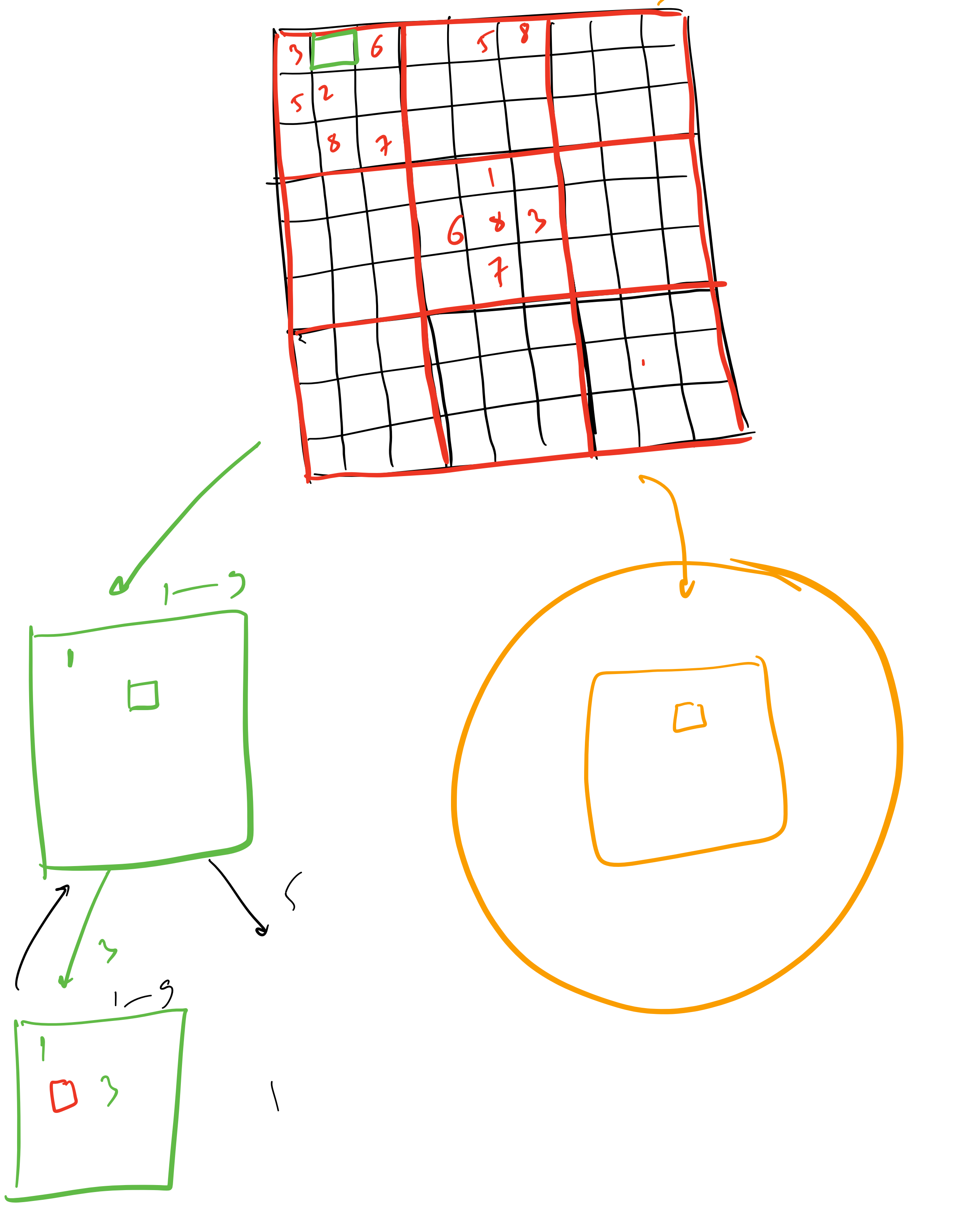
HS LU
HS RU
HS LL

ru

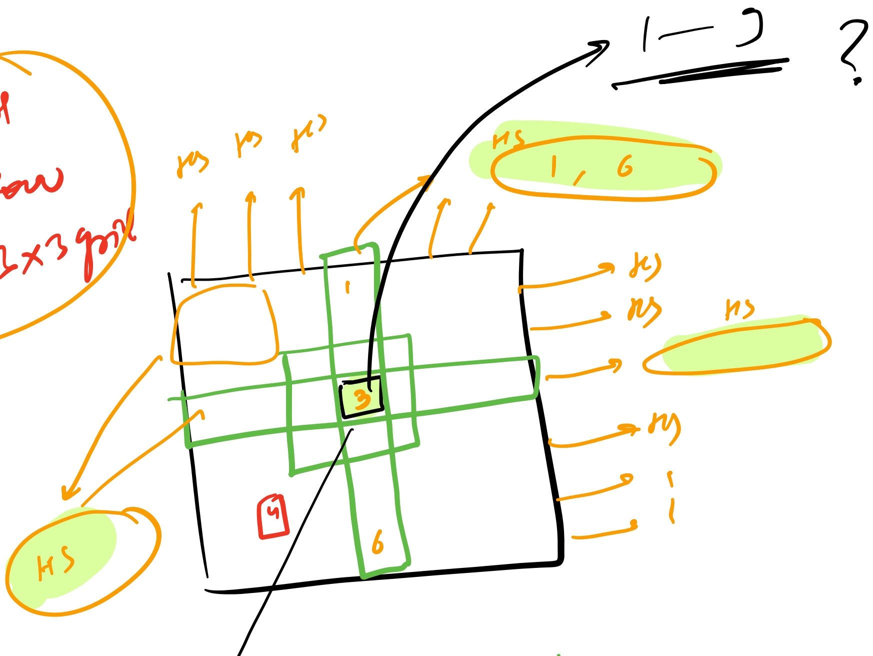
SUDOKU

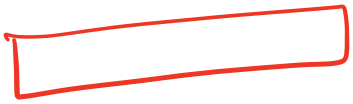
Row → {1, .. 9}
Col → {1, -- 9}
3x3 → {1 — 9}

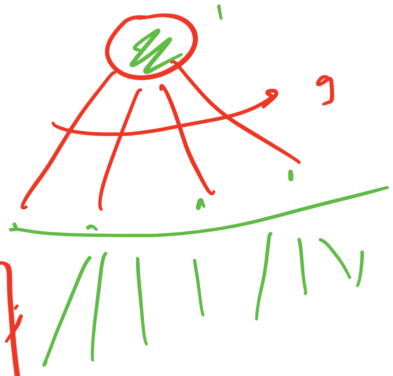
1 ?



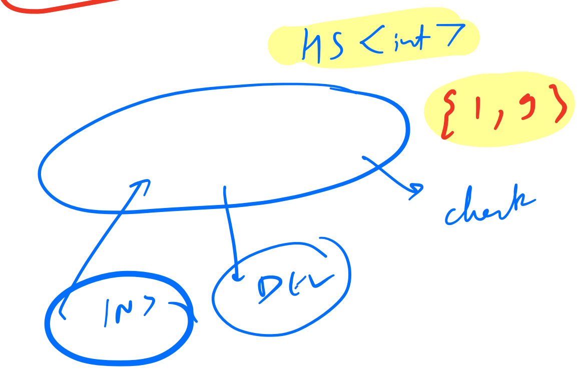
9 HS → Col
9 HS → Row
9 HS → 3x3 grid

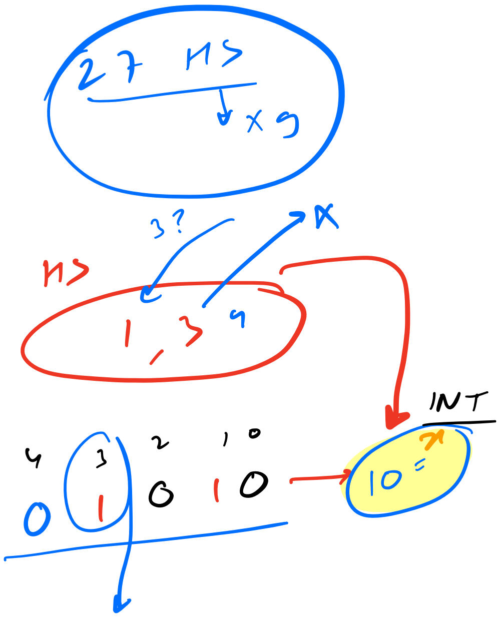






Hashset <int> hsr[9];





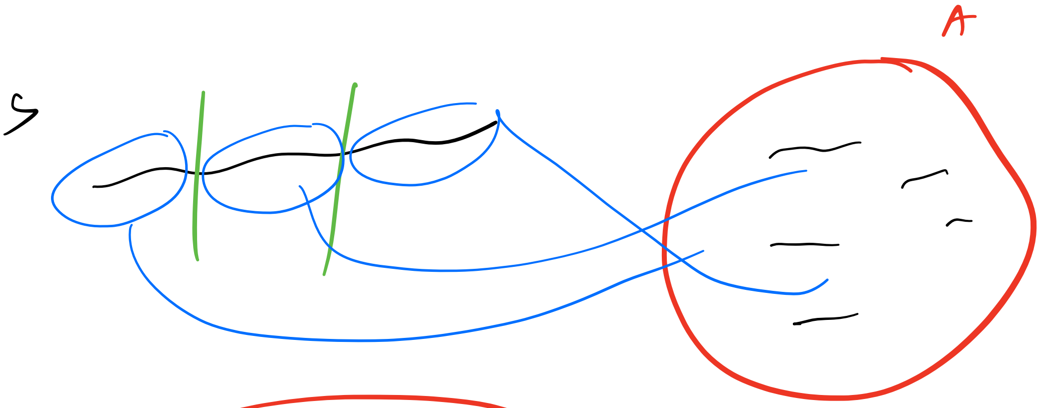
1 HS → 1 int

27 int

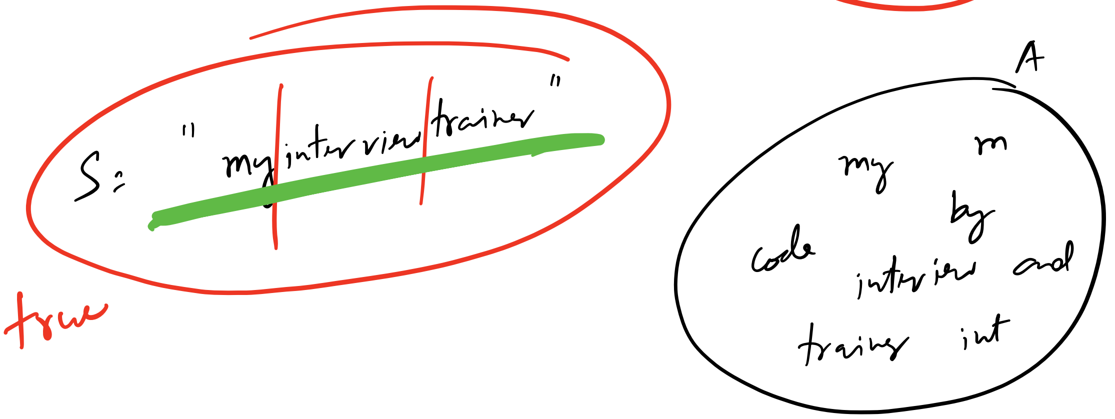
Word Break

Q

Given a string S & a set of words A.
Determine if S can be segmented into a
space separated sequence of one or more words
from A !



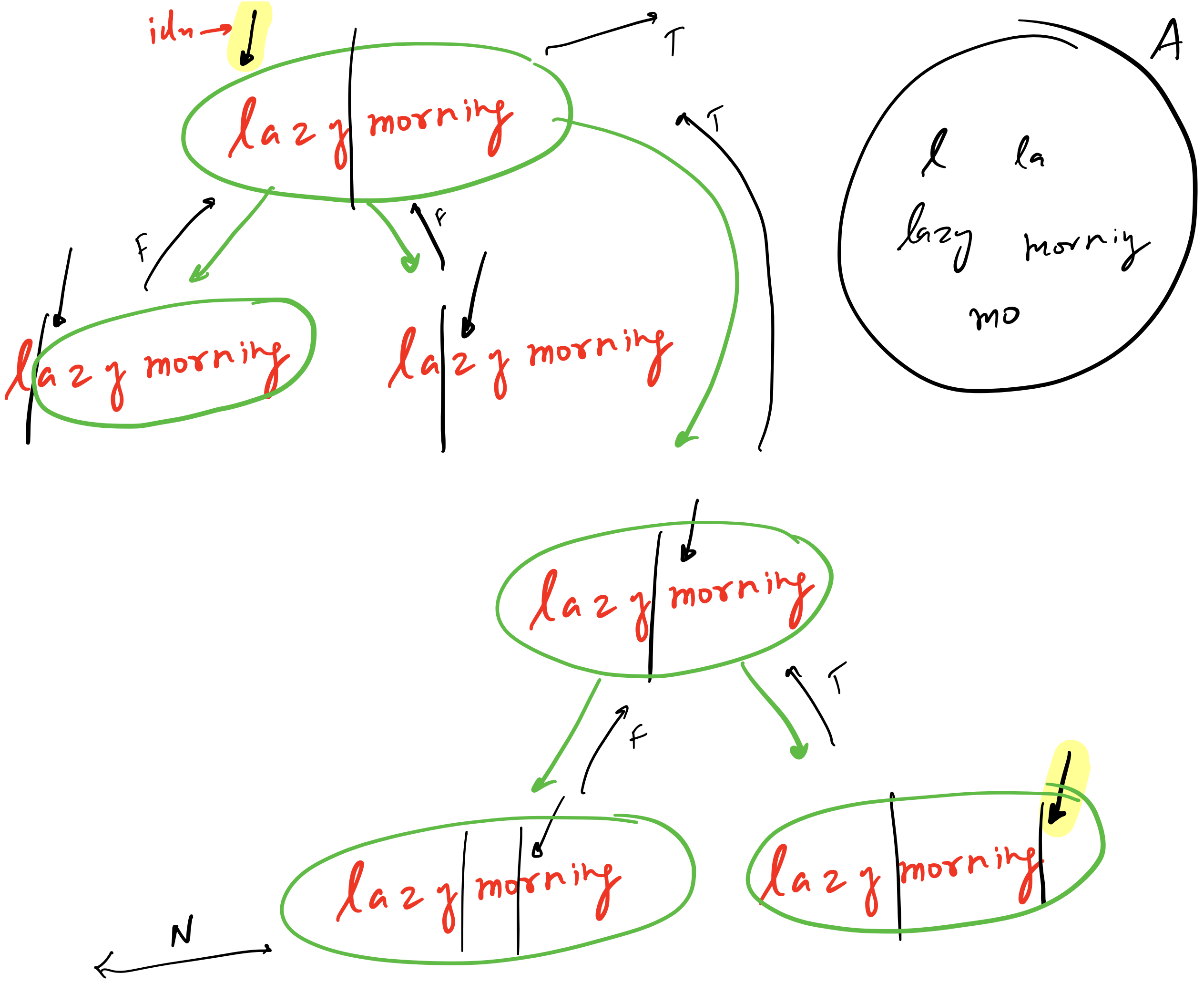
Eg

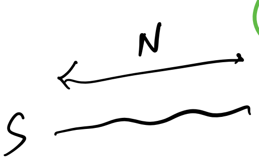


S: "my|interview|trainer"

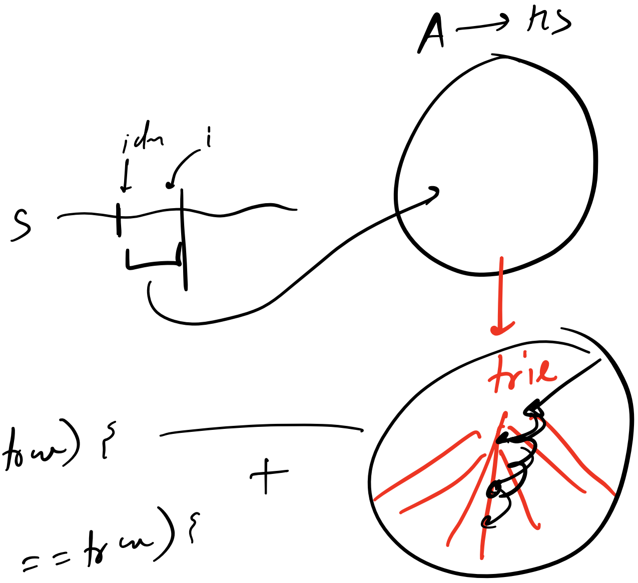
true

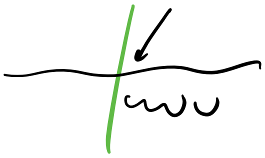
A: my, m, code, by, interview, and, trainer, int





```
bool check( s, idx) {
    if (idx == N) ret true;

    f( i = idx; i < N; i++) {
        ss = s.substring( idx, i);
        if( A.contains( ss) == true) {
            if (check( s, i+1) == true) {
                ret true;
    }
```





```
        }
    }
    ret false;
}
```