Q Given N & S.
Find the no. of N digit no's whose sum of digits = S.

$1 \le N \le 10^3$
$1 \le S \le 10^3$

N=2, S=4

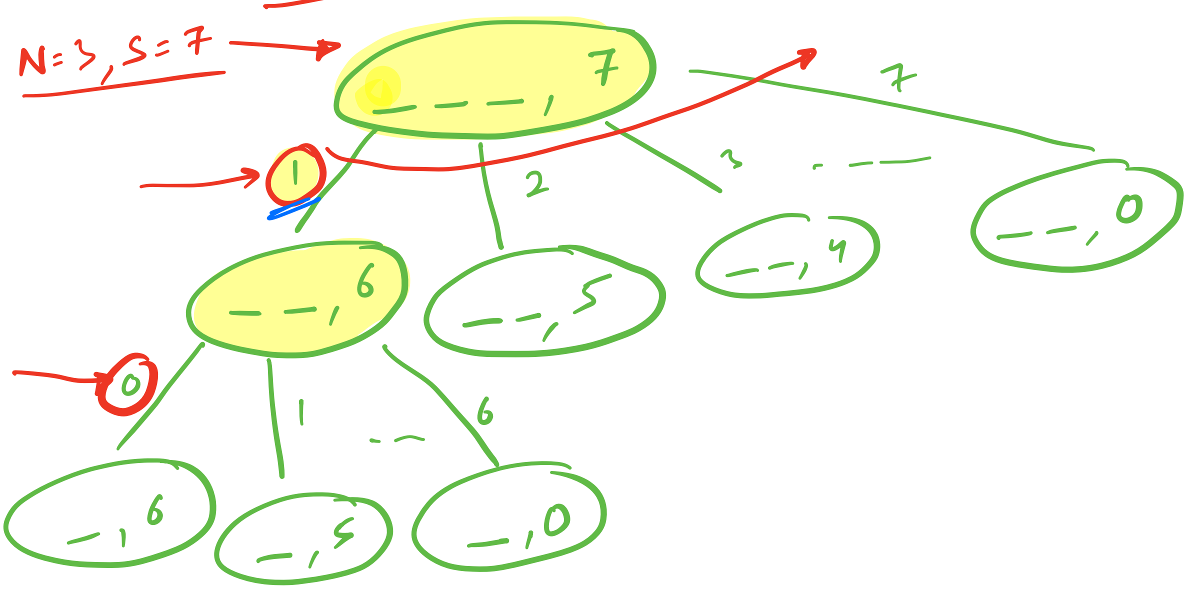
1	3
2	2
3	1
4	0

$\rightarrow$ 4

N=2, S=0 $\rightarrow$ 0

N=1, S=0   0 $\rightarrow$ 1

N=3, S=7



ways(n, s) = # of ways of making n digit no's whose sum of digits = s

ways(n, $\overset{6}{s}$)

- 0 → ways(n-1, s)
- \+ 1 → ways(n-1, s-1)
- \+ 2 → ways(n-1, s-2)
- \+ ⋮
- \+ 9 → ways(n-1, s-9)

RR

$$ways(n, s) = \sum_{k=0}^{k=\min(9,s)} ways(n-1, s-k)$$

n → [0 – N], s → [0 – S]

(N+1) × (S+1) → US → O(NS)

```
int dp[N+1][S+1] = {-1};

int ways(int n, int s) {
    if(n==0){
        if(s==0) ret 1;
        else ret 0;
    }
    if(s > n×9) ret 0;
    if(dp[n][s] != -1){
        ret dp[n][S];
    }
    int ans = 0;
    f(k=0; k <= min(9,s); k++){
        ans += ways(n-1, S-k);
    }
    dp[n][s] = ans;
    ret ans;
}
```

BC ?

HW

N≥1

% $10^9 + 7$

MAIN()

```
if(S==0){
    if(N>1)
        ret 0;
    else ret 1;
}

ANS = 0

f(i = 1 → min(9,S))
    ANS += ways(N-1, S-1)
```

TC per state → 10 ops

O(1)

$TC = \#VS \times TRPS$

$= NS \times O(1)$ <10>

$$\boxed{TC = O(\overset{10}{NS})}$$

$S <= 9.N$

$N \times 9 \times N$

$$\boxed{TC = O(N^2)}$$

$SC = \#VS$

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

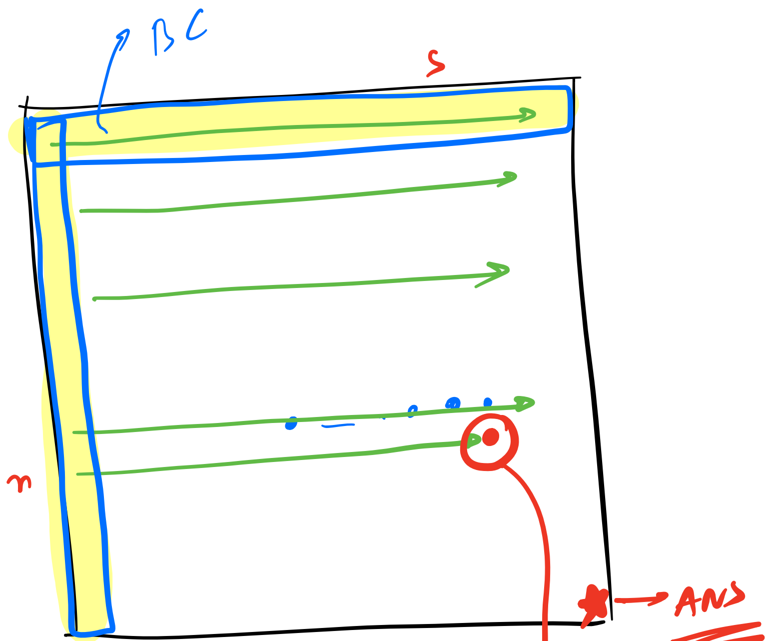
Bottom Up Analysis

~~RR~~

$$way(n, s) = \sum_{k=0}^{k=\min(9,s)} way(n-1, s-k)$$

N, S

way(N, S)

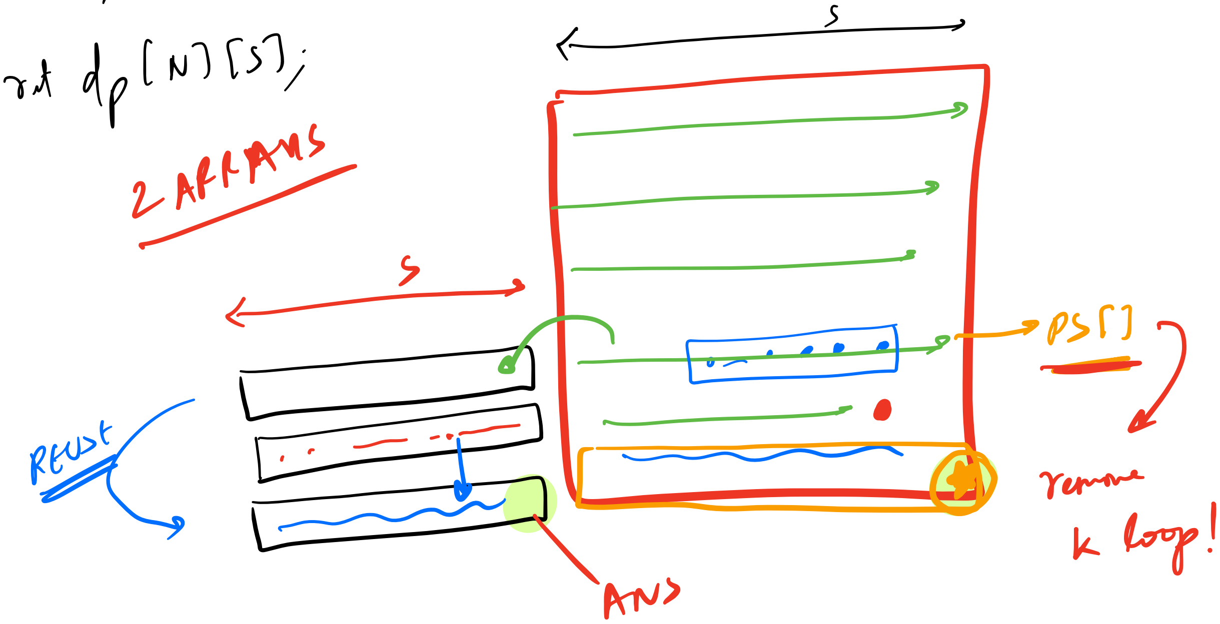


n, s

```
f(n=0; n<=N; n++){
    f(s=0; s<=S; s++){
        ANS=0
        f(k=0; k<=min(9,s); k++){
            ANS += dp[n-1][s-k];
        }
        dp[n][s] = ANS;
}
```

}

int dp[N][S];



$\boxed{SC = O(S)}$ $\boxed{TC = O(NS)}$

Q Given a 2D binary matrix of N×M. A[i][j] ∈ {0, 1}

start (0,0)

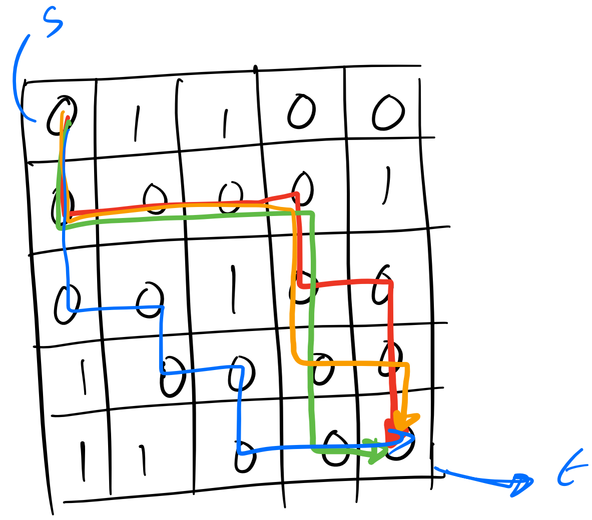
end (N-1, M-1)

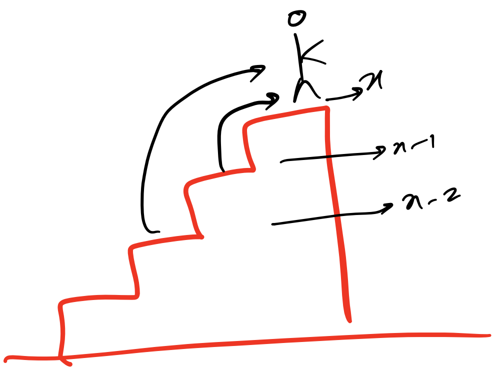
find the # of ways of reaching end from start!

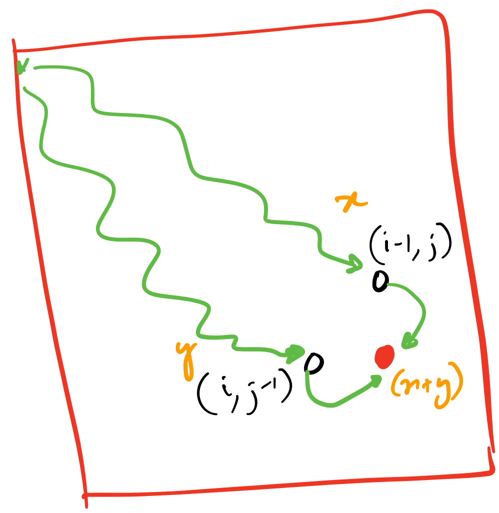
NOTE:

1. Go right or down

2. 1 : Obstacle ✗
   0 : Empty space. ✓







$\boxed{ways(i, j) \rightarrow \text{\# ways of reaching } (i, j)^{th} \text{ cell from } (0, 0)^{th} \text{ cell.}}$

RR

$\boxed{ways(i, j) = ways(i-1, j) + ways(i, j-1)}$ $\because A_{ij} = 0$

$\rightarrow 0$ $\because A_{ij} = 1$

[0 – N-1] [0 – M-1]

N x M

```
int dp[N][M] = {-1};

int ways(int i, int j) {
    if(i<0 || j<0) ret 0;
    if(A[i][j]==1) ret 0;
    if(i==0 && j==0) ret 1;
    if( dp[i][j] != -1) {
        ret dp[i][j];
    }
    ans = ways(i-1, j) + ways(i, j-1);
    dp[i][j] = ans;
    ret ans;
}
```

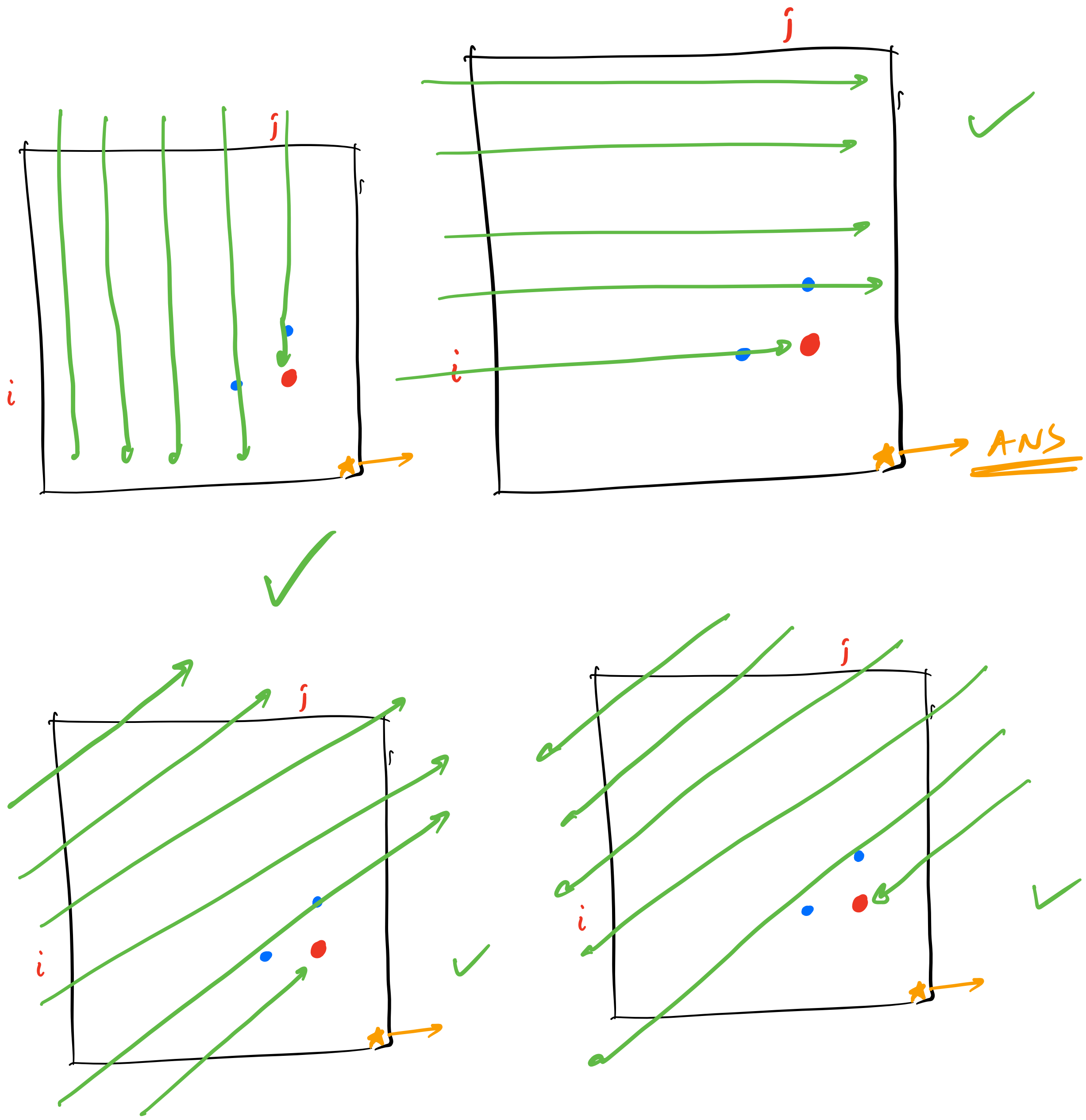
#US → NM

TRPS → O(1)

$\boxed{TC = O(NM)}$

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

Bottom Up Analysis



f(i = ~~0~~ 1 → N-1) {

f(j = ~~0~~ 1 → M-1) {

// (i, j)

→ if(A(i, j) == 1) { dp[i][j] = 0; continue; }

dp[i][j] = dp[i-1][j] + dp[i][j-1]

<0 <0

}

}

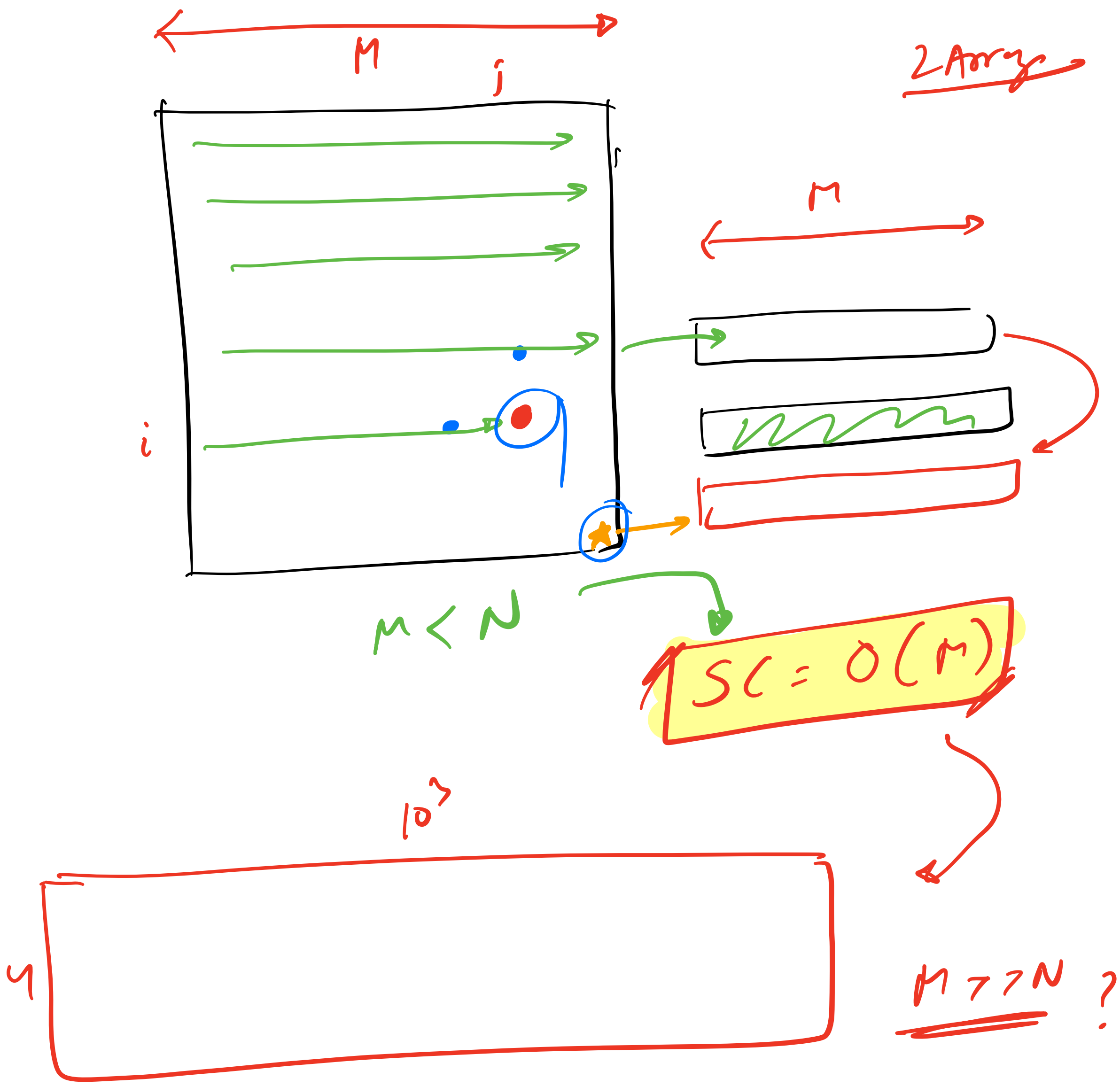
$\boxed{TC = O(NM)}$ ✓

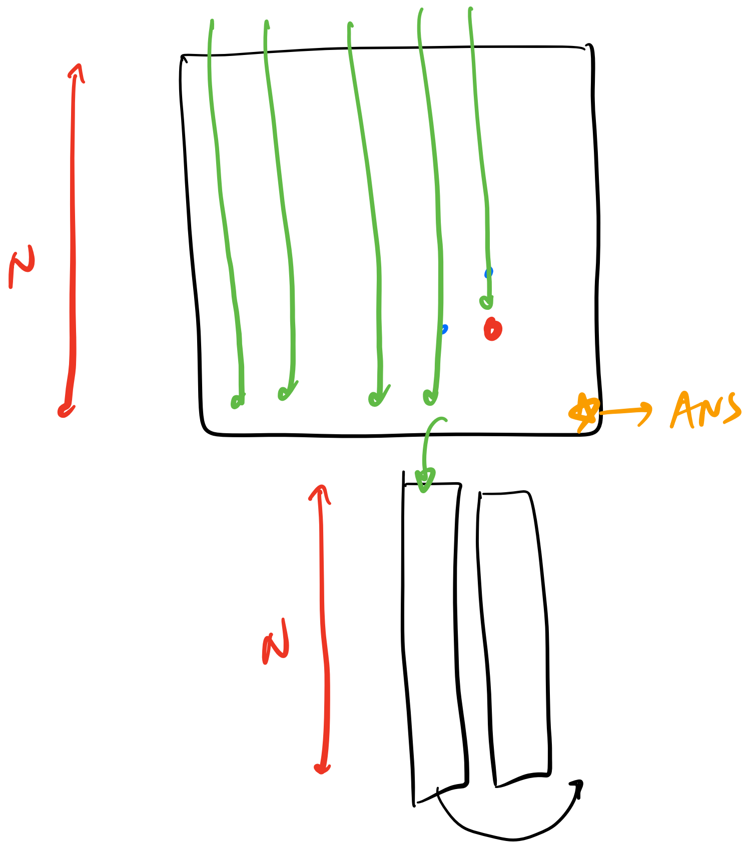
0	1	1	0	0
0	0	0	0	1
0	0	1	0	0
1	0	0	0	0
1	1	0	0	0

1	0	0	0	0
1				
1				
0				
0				

KR

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

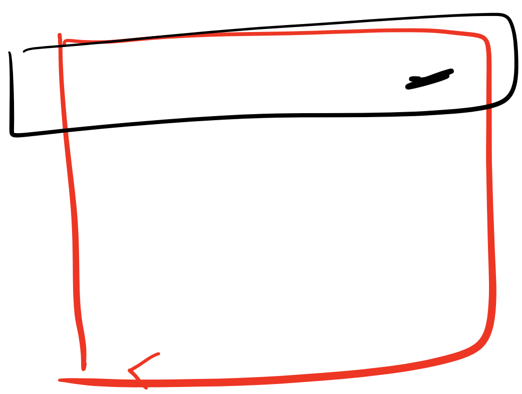

2 Arrays
M
j
i
M
M < N
SC = O(M)
10^3
4
M >> N ?



$N < M$

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

$$\boxed{SC = O(\min(N, M))}$$



DOUBT