



# **CIPHERX CAPSTONE**

**P9 TV SHOW SCHEDULER**

# CONTENT

- 01** OUT TEAM
- 02** PROBLEM
- 03** ALGORITHM
- 04** DATA STRUCTURE CHOSEN
- 05** TIME COMPLEXITY
- 06** SPACE COMPLEXITY
- 07** VIDEO

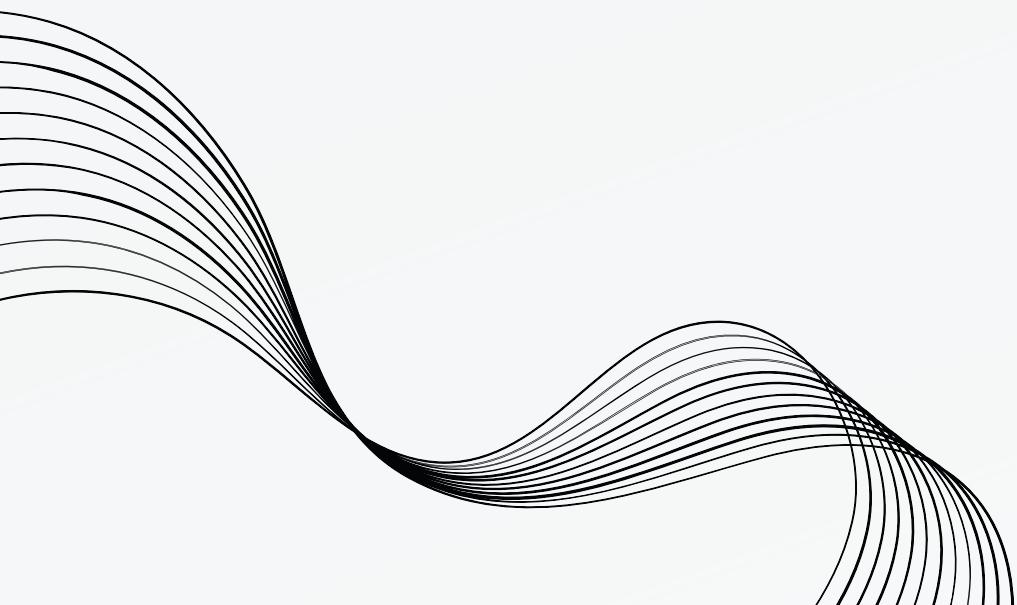
# OUR TEAM

Manan Chhabhaya 202301222

Kresha Vora 202301231

Anushka Prajapati 202301224

Kashyap Ajudiya 202301239



# PROBLEM

P9

We have given a text file containing M Distinct Series, and N family members, each having a favorite series and Availability slots. We have to schedule their favorite series into their availability slots such as The Scheduler needs to minimize total episode misses. There can be the same favorite.

We have tried to solve this problem as efficiently as we can. We have also given option to show personal schedule of any person user wanted to get. User can get multiple file to schedule their TV series

# TIME COMPLEXITY

N = number of member

M = number of series

Assumed total time slots (T) = 168 (24\*7)

Time complexity of functions :

Input() :  $O(N \cdot M)$

storeSlotVsSeries() :  $O(N \cdot M)$

findMax() :  $O(M \cdot N)$

Assumed that The number of series M <= 168 , as number of series cannot exceed number of slots.

Overall Time complexity :  $O(N)$  , with assumptions

$O(N \cdot M)$  , M - variable( $M > 168$ )

# SPACE COMPLEXITY

N = number of member

M = number of series

Space complexity of arrays created into the code:

memVsSlot :  $O(N)$

memVsSeries :  $O(M*N)$

slotVsSeries :  $O(M)$

Overall Space complexity :  $O(N)$  , with assumptions

$O(N*M)$  , M - variable( $M > 168$ )

# DATA STRUCTURE CHOSEN

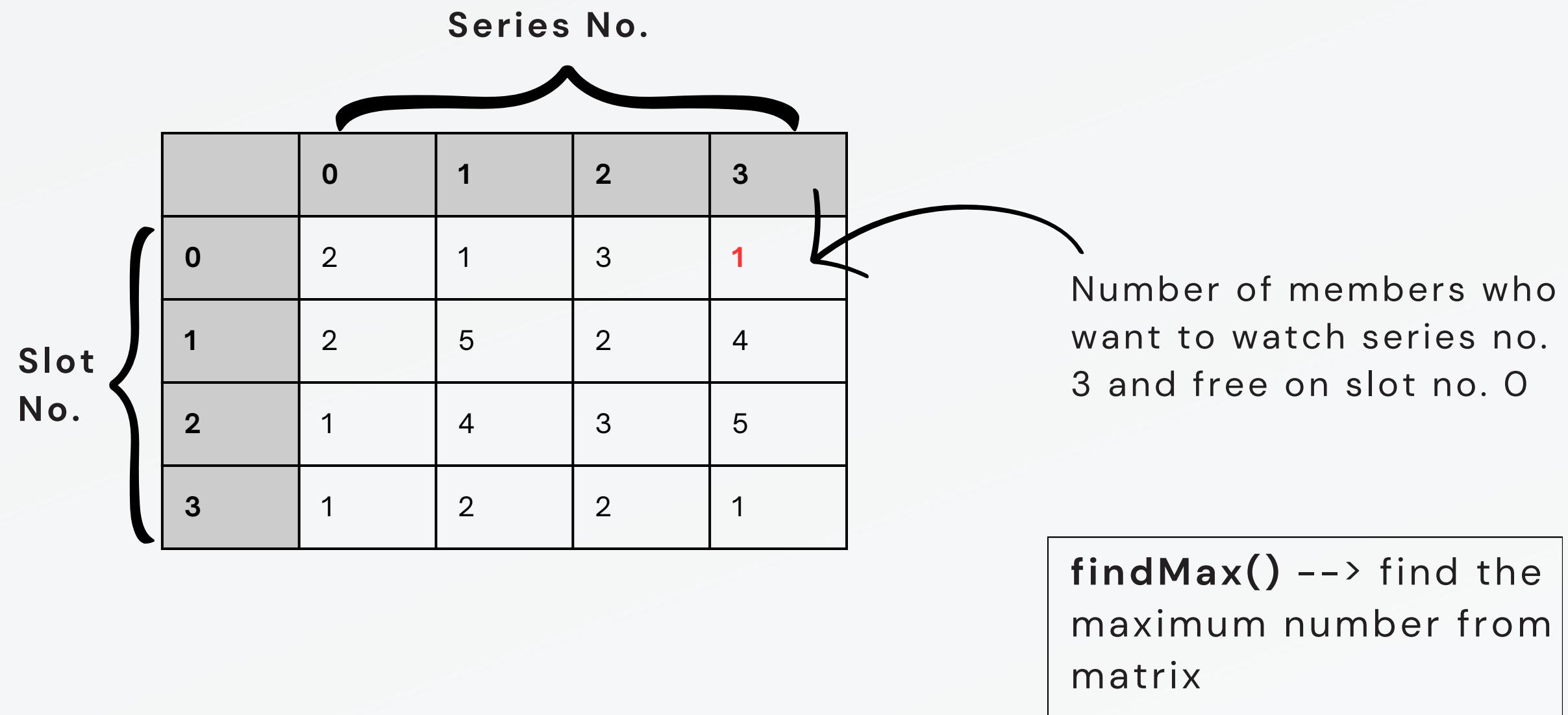
## Vectors

- Vectors are more efficient in insertion and deletion of data.
- It also provides indexing like arrays. but also provides dynamic data storing like linked list.

## Priority Queue

- Additionally, we have implemented priority queue to store series in slot wise order.

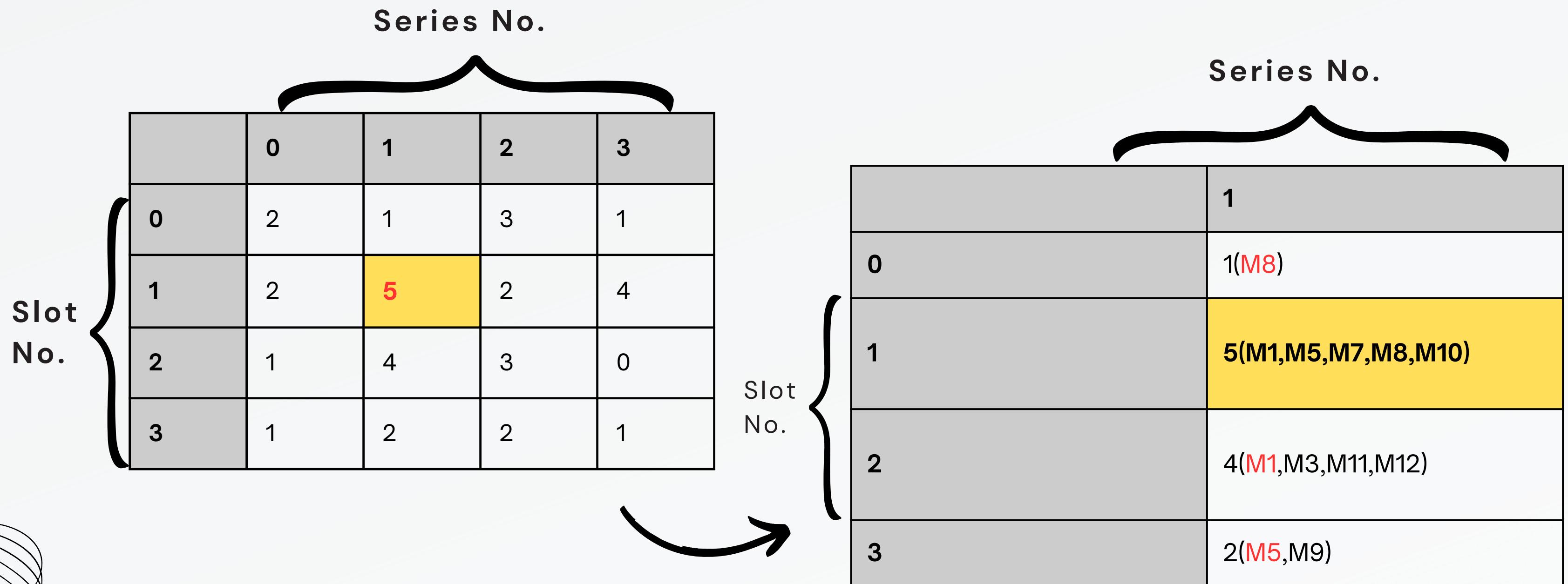
# ALGORITHM



# ALGORITHM

CASE 1

Only one maximum number



# ALGORITHM

CASE 2

Two same max number in matrix

	0	1	2	3
0	2	1	3	1
1	2	5	2	4
2	1	4	3	5
3	1	2	2	1

	1	3
0	1(M8)	1(M4)
1	5(M1,M5,M7,M8, M10)	4(M2,M9,M10,M11)
2	4(M1,M3,M11,M12 )	5(M2,M4,M9,M11, M12)
3	2(M5,M9)	1(M4)

# ALGORITHM

CASE 2

		Series No.
		1                    3
Slot No.		
0	1(M8)	1(M4)
1	5(M1,M5,M7,M8, M10)	4(M2,M9,M10,M11)
2	4(M1,M3,M11,M12 )	5(M2,M4,M9,M11, M12)
3	2(M5,M9)	1(M4)

Whenever we find two same max number in matrix

FOR SERIES 1 :

The number of common members free in other slots for series 1 (A) --> 3

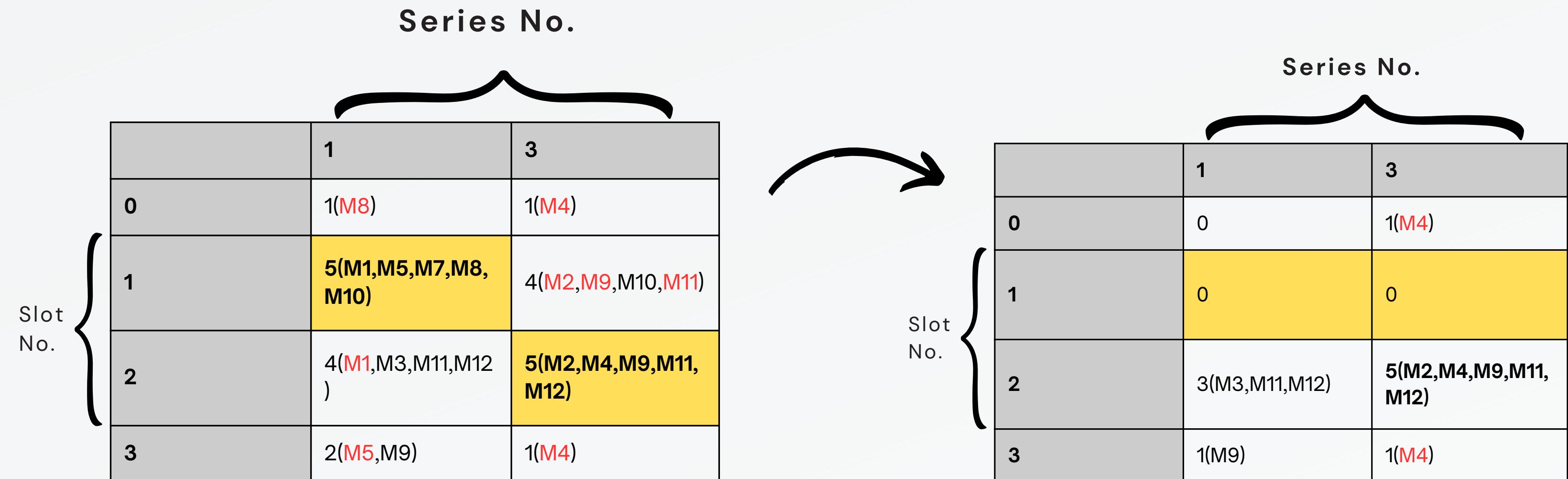
FOR SERIES 3 :

The number of common members free in other slots for series 1 (B)--> 5

The greater this number the greater opportunities to watch that series in other slots.

so we should choose slot 1 because these members have less opportunity

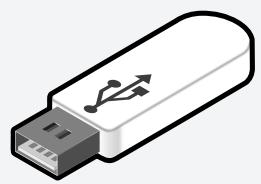
# ALGORITHM



after finalizing slot. Deduct all common member from that column and change all value of slot 1 to 0.

DO this until all values convert into 0.

# RECORDING



DRIVE LINK

# THANK YOU

