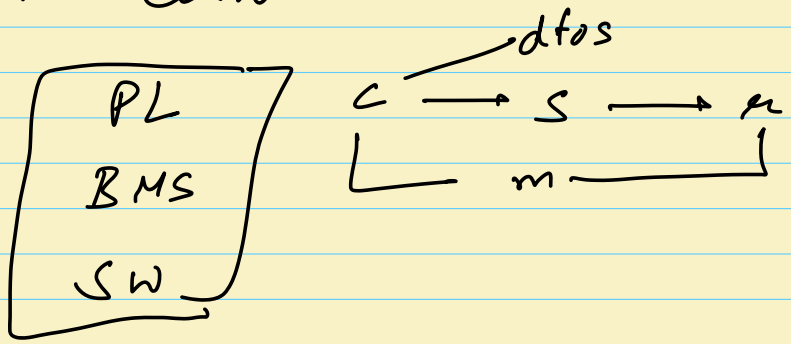
1. Good Evening
2. Lecture begins at 9:10pm
3. Topic → Splitwise
   Pending TTT

Agenda

1. Command Design Pattern
2. Restful API → Postman
3. One data flow – end to end.
4. TTT Undo Strategy
5. TTT WS "

Command Design Pattern.

→ TTT Client

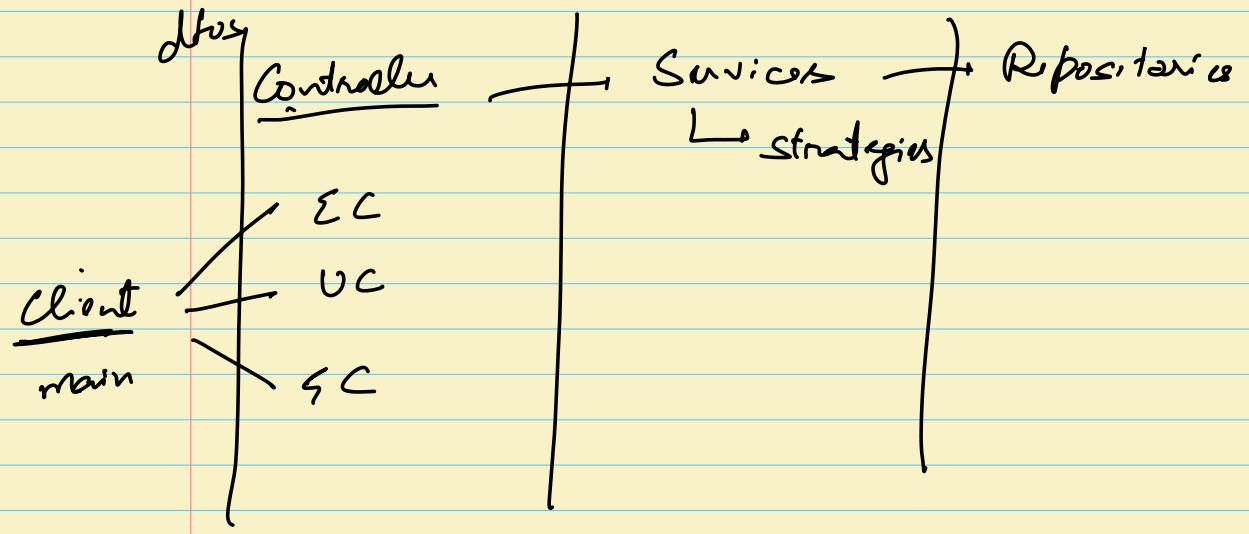


Splitwise

→ Expense Controller → createExpense

→ User Controller → createUser

→ Group Controller → SGRDTO → SetToGroup(SGRDTO)



```
while (true) {
    String input = scn.nextLine()
    if (input.sw("CreateExpense") {
        ExpenseController ec = ——
        ec.createExpense()
    } else if (input.sw("settleGroup") {
        GroupController gc = ——
        gc.settleGroup(——);
    }
}
```

settleGroup to

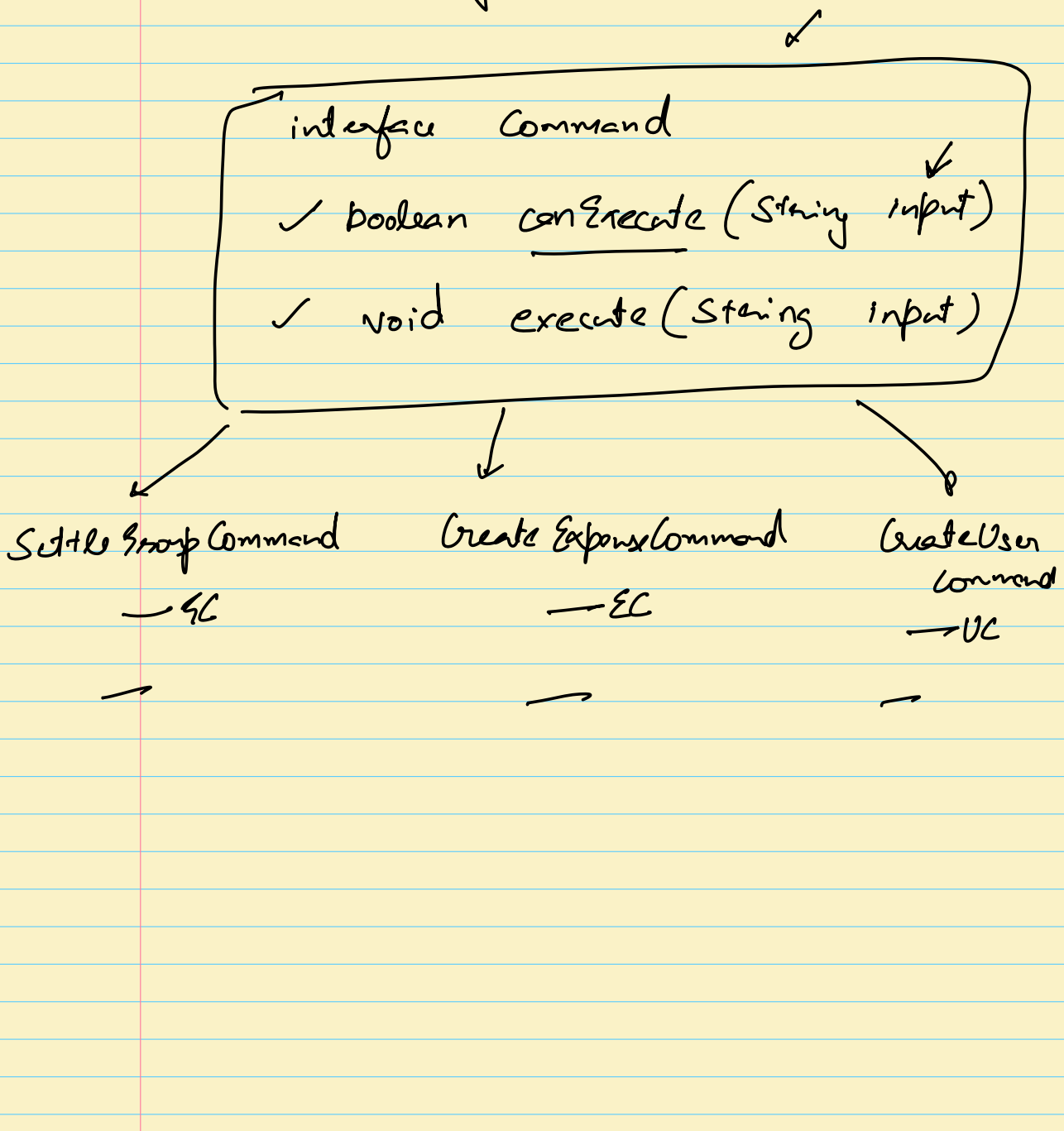
createExpense ——————————

```
if(——) {
    ✓
} else if( ) {
    ✓
} else if ( ) {
    ✓
}
```

SRP ✗ OCP

cf ( )

Command Design Pattern.



@Controller
UserController
@Autowired
UserService

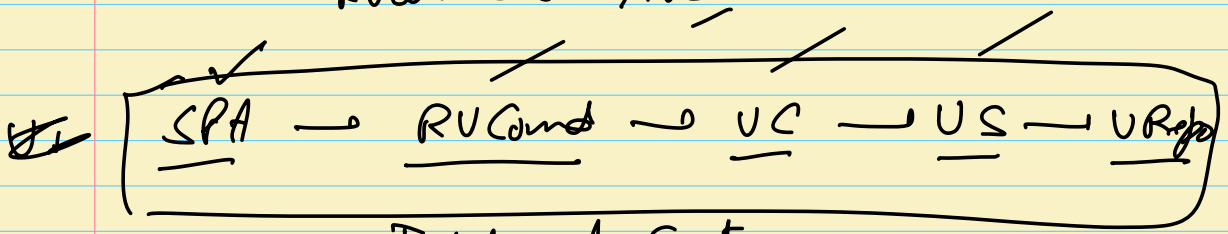
@Service
UserService
@Autowired
UserRepo.

@Repo
UserRepo

@Component
RUCommand
@Autowired
UserController.uc

Graph of it

@SpringApplication
SPA
@Autowired
RUCommand ruc;



Topological Sort

URepo.

Agenda

1. Command Design Pattern ✓

2. Restful API → 5 steps
3. Data flow = end to end

Create a new user

End details of a user

4. TTT Undo
5. TTT WS O(1)

10:28 to 10:38 Break

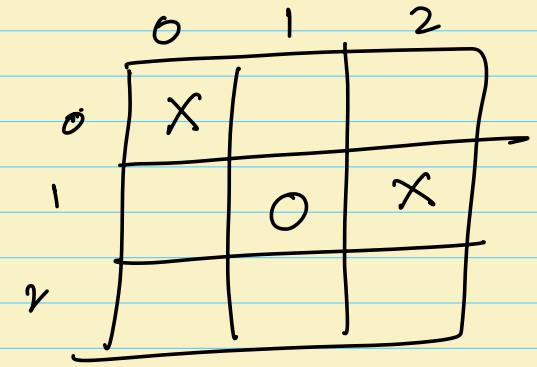
1. Command ✓

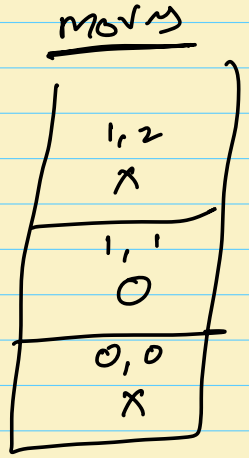
2. Restful API ✓
3. Data flows ✓

4. TTT Undo
5. TTT O(1) WS.

TTT

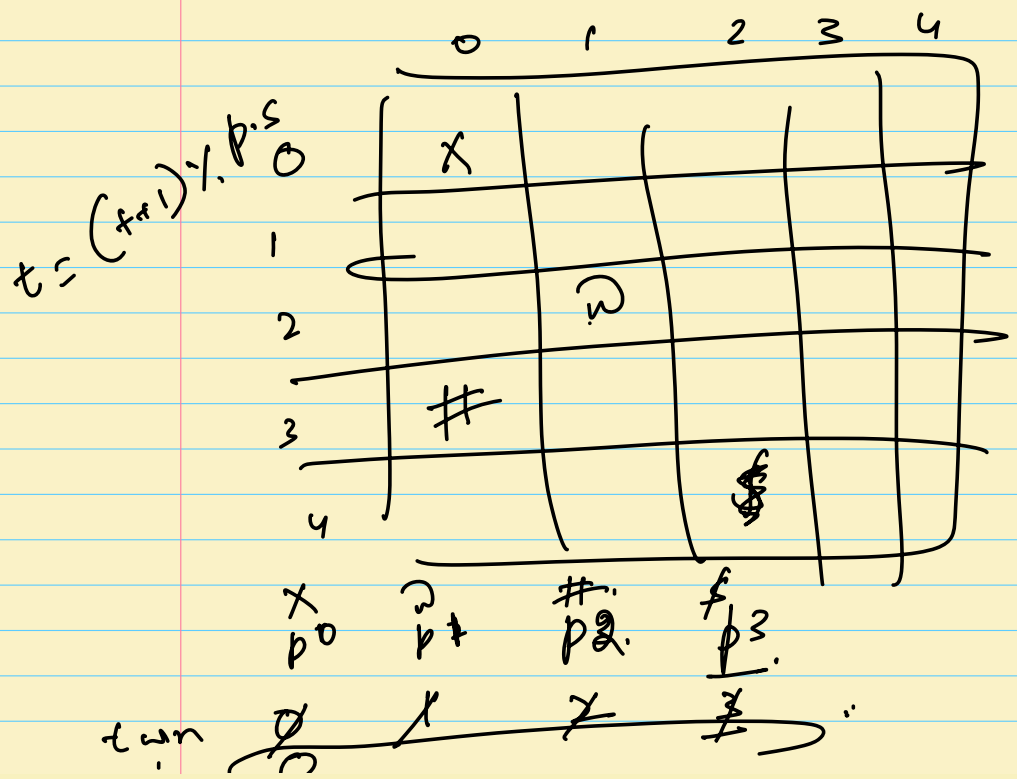
1. Undo

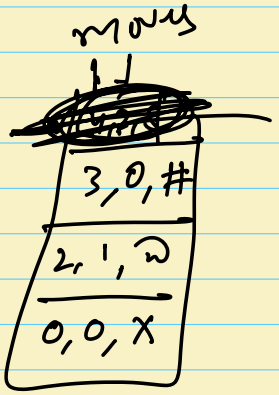




players = 2

turn = 0, 1





p^0 p^1 p^2 p^3

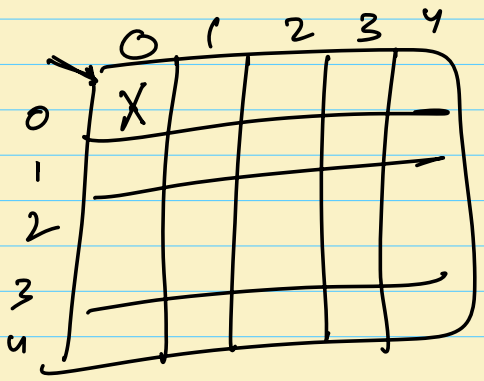
$\emptyset$ x z 3

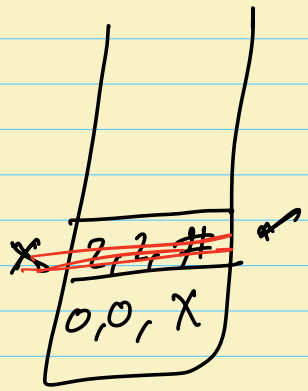
X # $ ☆

p^0 p^1 p^2 p^3

2

turn $\emptyset$ 1





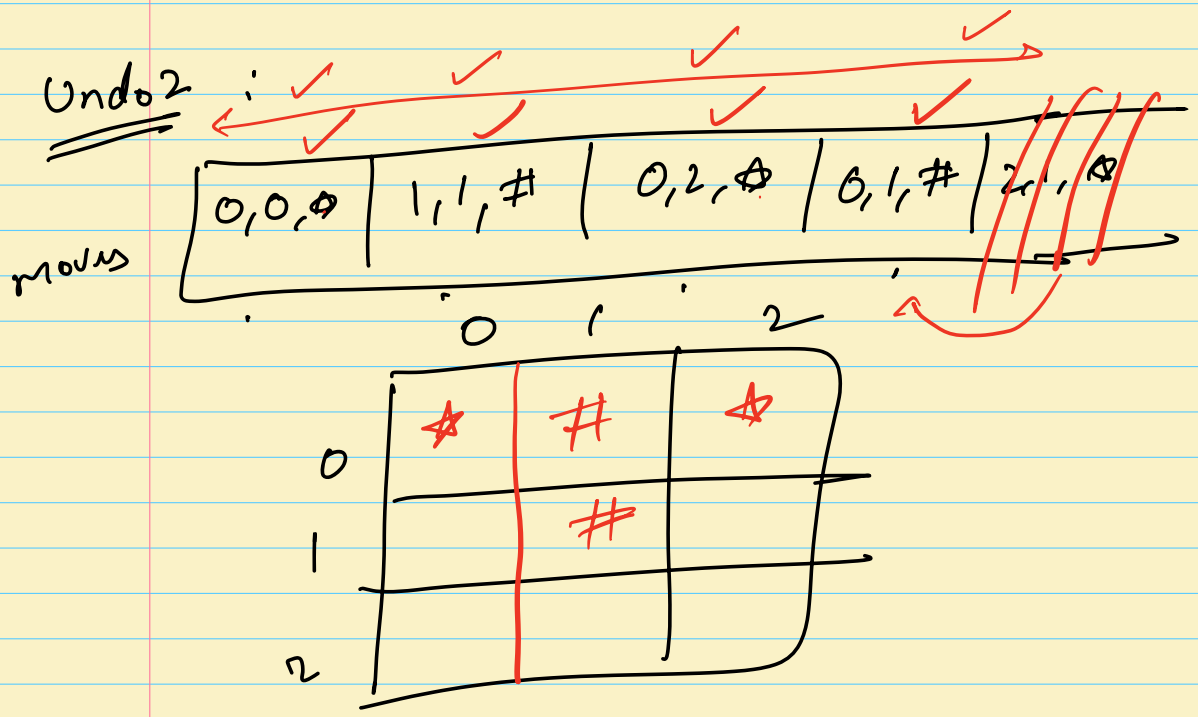
moves

2, 2, #

Undo 1 → Make necessary changes

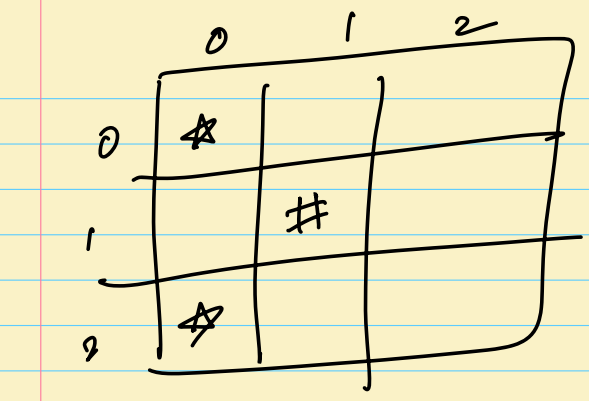
1. Clear the board of last move
2. Change turn value
3. Pop from moves.

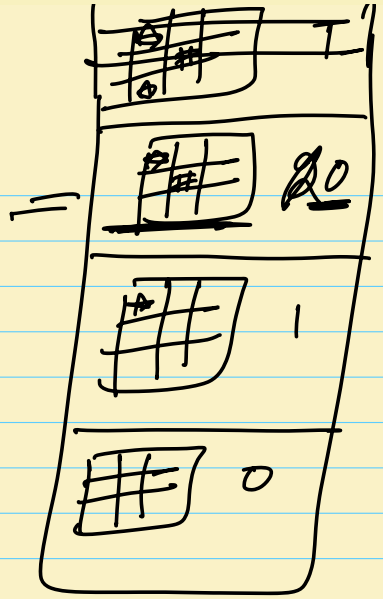
Undo 2 :



→ inefficient on time.

Undo 3 :





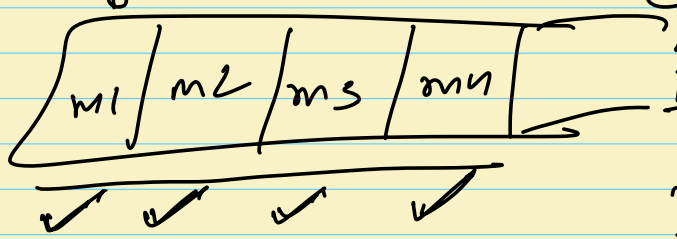
turn = 0 1

P0 P1

☆ #

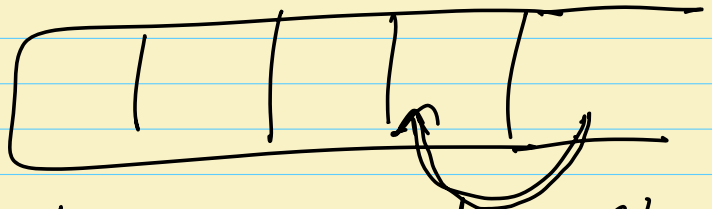
Undo 3 → Saving gaming history → Bad on memory, Fast on RT

Undo 2 → 0



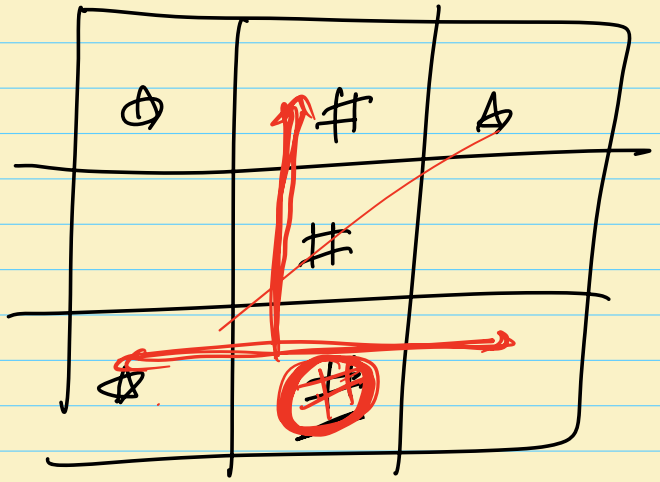
Good on memory, bad on TC

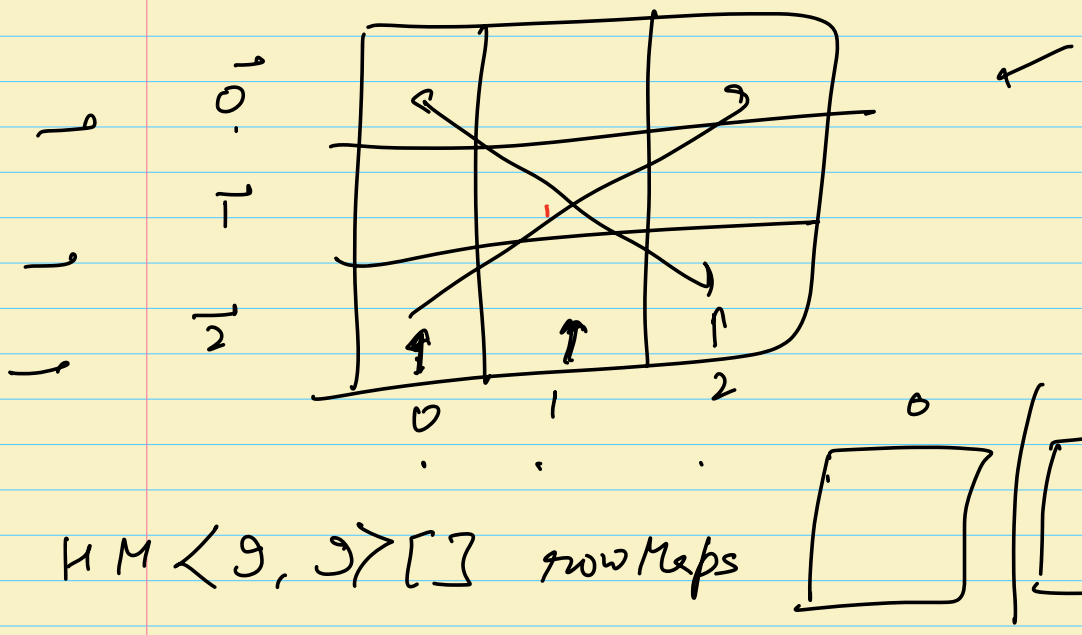
✓ Undo 1 →



Good on TC, Good on Space.

O(1) ws



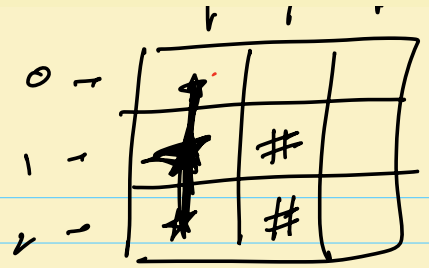


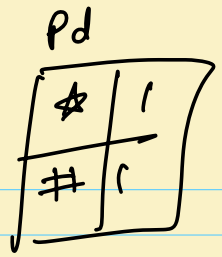
HM<9, 9>[] rowMaps

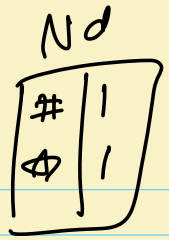
HM<9', 9>[] colMaps

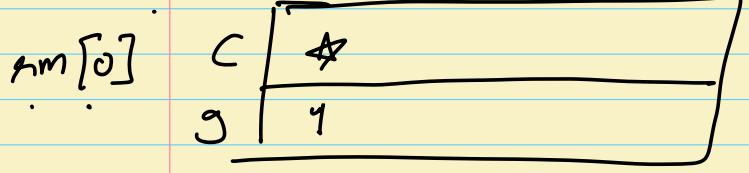
HM<9.9> posdia

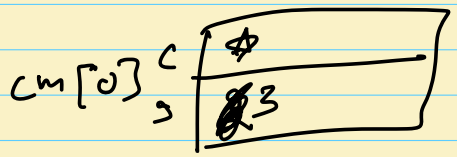
HM<9,9> negdia
0 1, 2

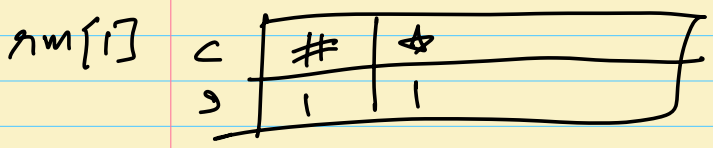


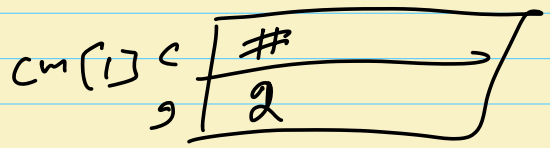


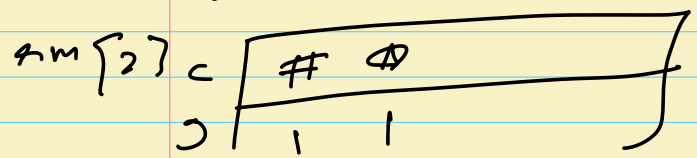


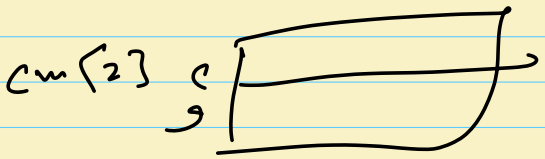










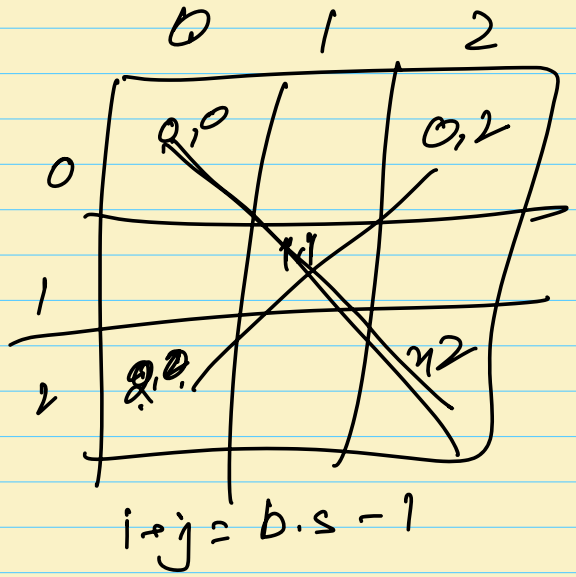


HM<C,g> pd map

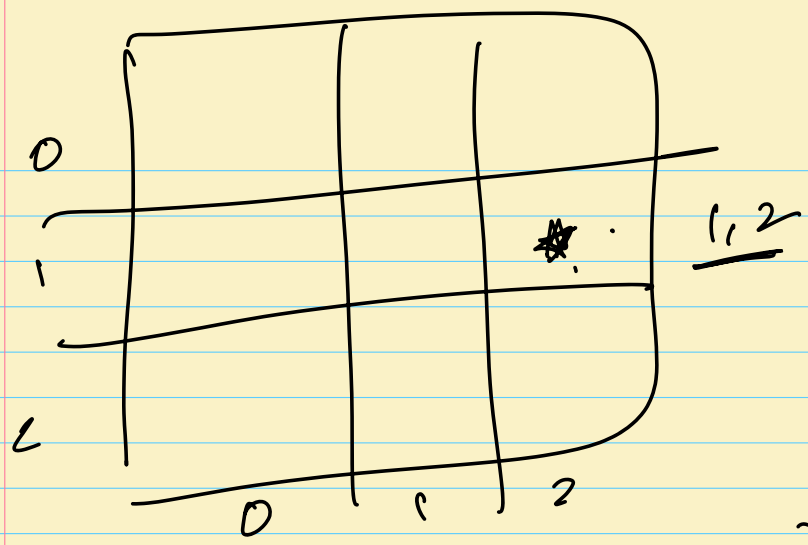
|| nd map

HM<C,g>[] row maps

|| col maps



i+j = b.s - 1



arr[i]

cur[2]

if (r == c)
  pr —

if (r+c == b.s-1)
  rm —

SP
/
Common DP ✓

✓ Restful API
✓ Data Flow

[Undo TTT ✓
  TTT O(1) WS ✓]

Leetcode → (500) ✓ / Scaler

Code Forces → 1600 ~~2000~~ ~~1800~~ ✓ (DP)

10, 20

2000

DP, Graph, BT, Ans