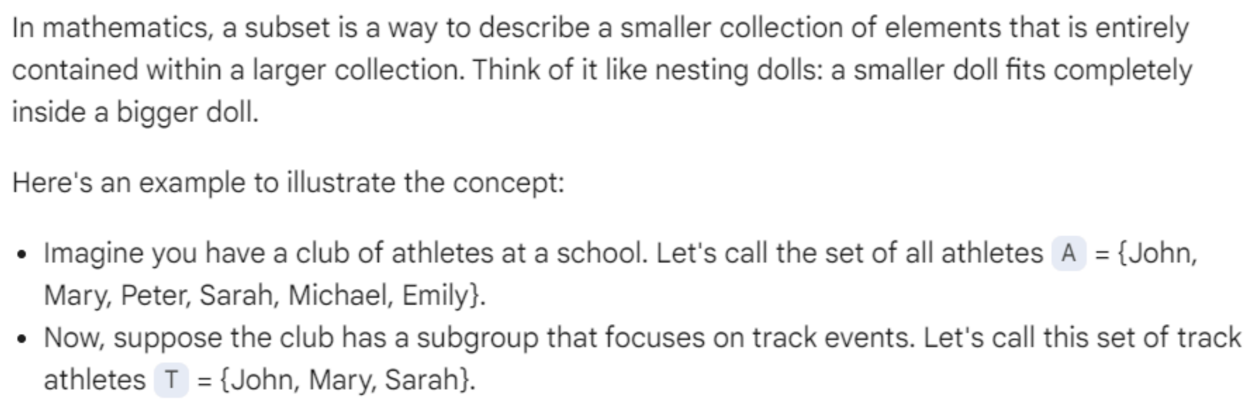
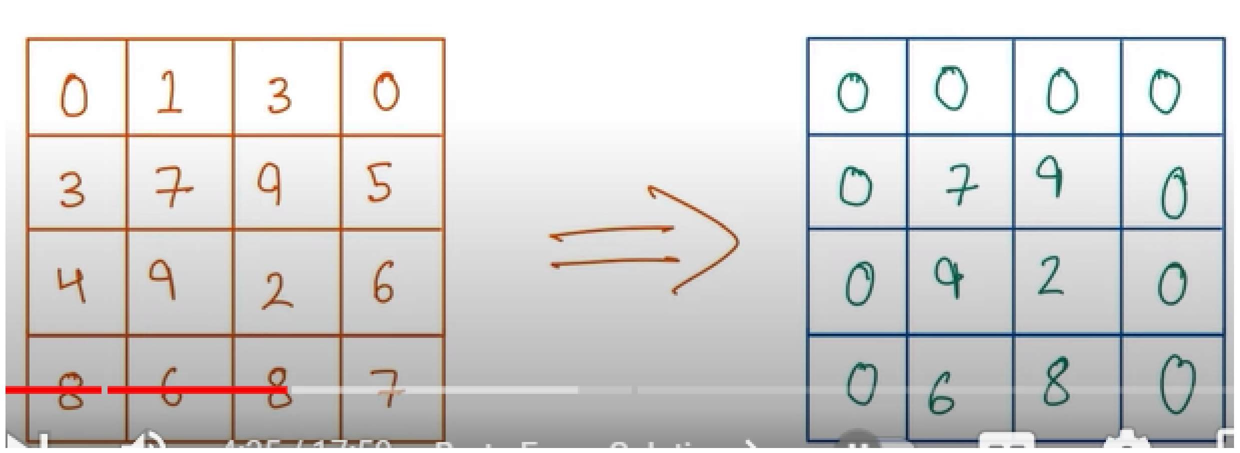
Keyboard_row_leetcode

1. Lets consider two sets
   a1 = set('asdfghjkl')
   b1 = set('dadz')
2. Now if we want to know whether..we can print b1 with the characters present in a1..then we can use the concept of subset
   
   In mathematics, a subset is a way to describe a smaller collection of elements that is entirely contained within a larger collection. Think of it like nesting dolls: a smaller doll fits completely inside a bigger doll.

   Here's an example to illustrate the concept:

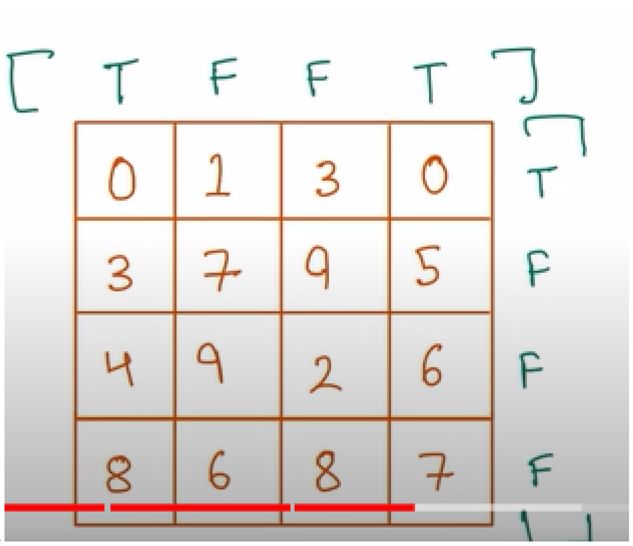
   - Imagine you have a club of athletes at a school. Let's call the set of all athletes A = {John, Mary, Peter, Sarah, Michael, Emily}.
   - Now, suppose the club has a subgroup that focuses on track events. Let's call this set of track athletes T = {John, Mary, Sarah}.
3.
4. Related problem : https://leetcode.com/problems/keyboard-row/description/

73. Set Matrix Zeroes

1. So in normal approach we just iterate rows and if their's a zero then will make entire row as 0..similarly for columns..will iterate every column and if there's a zero ..will make entire cols as 0
2. In this approach for every element..we have to iterate both row and col(which takes a lot of time) …this is a brute force sol
   

3. Now lets start optimizing this

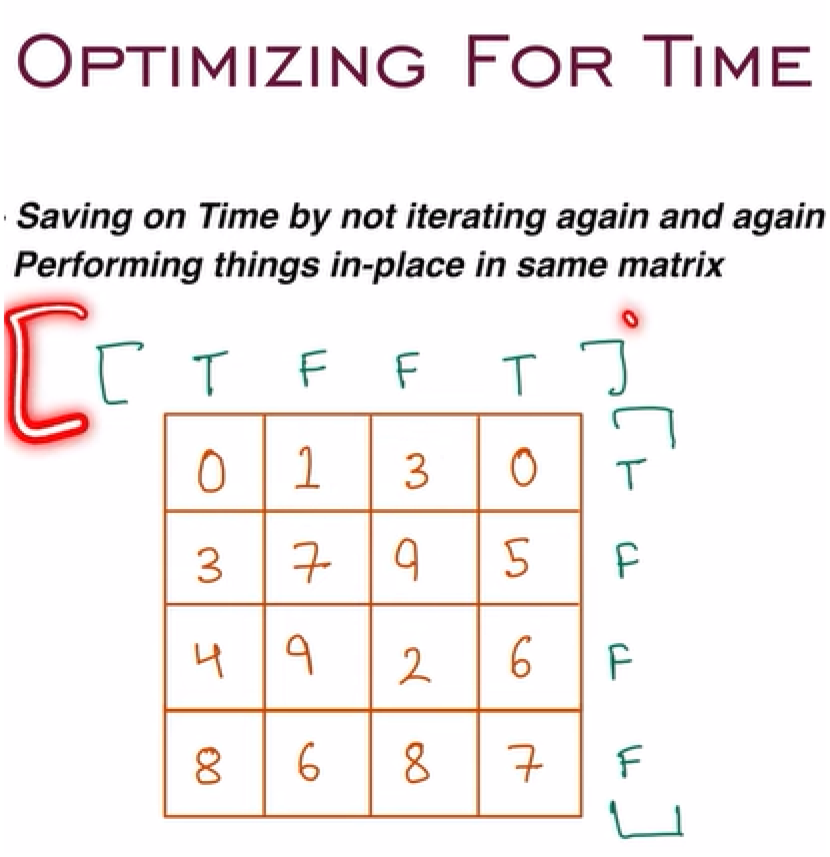
4. In this approach..we'll take 2 temporary lists…which is used to know the whether a row



or a column has 0 or not

5. So here if there's a 0 in the row ..we will mark it as 'T" else 'F'...similary for the column..if there's a zero in the column ..we will mark it as 'T' else 'F'

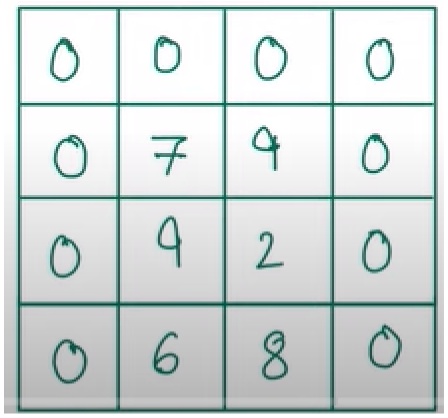
6. So now while iterating if there’s one true in col or row..then we’ll make that element 0



Optimizing For Time

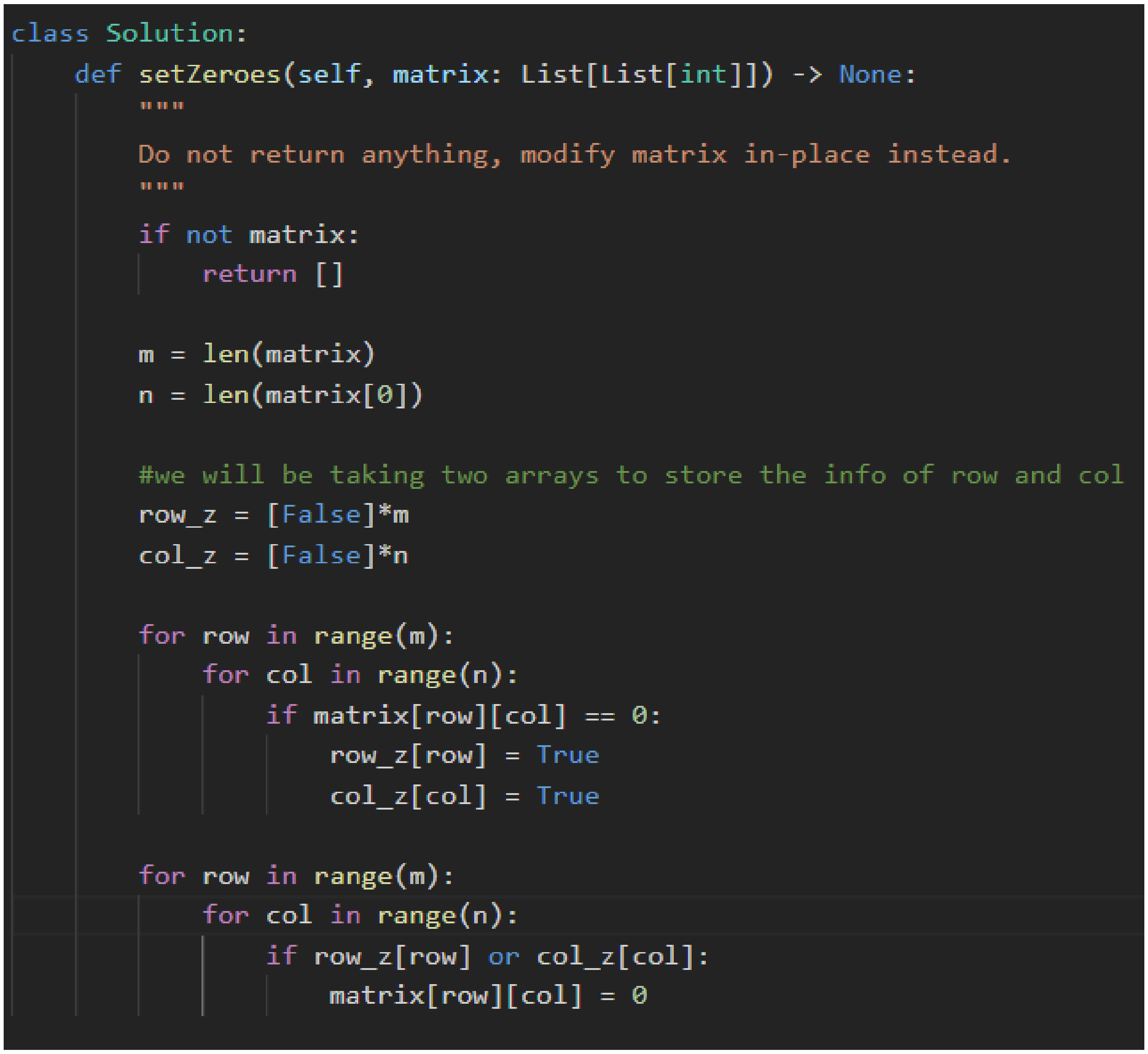
- Saving on Time by not iterating again and again
- Performing things in-place in same matrix

$$\begin{array}{c|cccc} & T & F & F & T \\ \hline T & 0 & 1 & 3 & 0 \\ F & 3 & 7 & 9 & 5 \\ F & 4 & 9 & 2 & 6 \\ F & 8 & 6 & 8 & 7 \end{array}$$

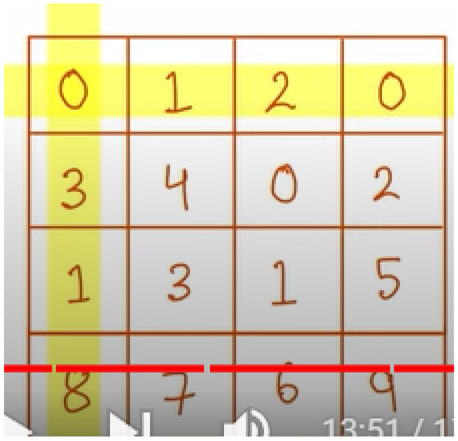


0	0	0	0
0	7	9	0
0	9	2	0
0	6	8	0

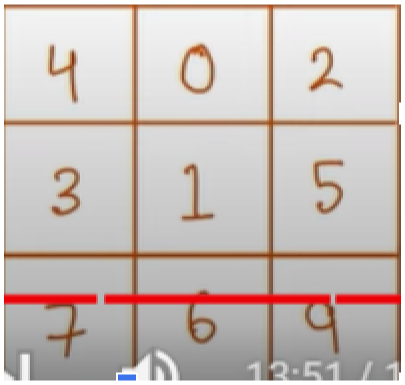
7. Code for the above approach



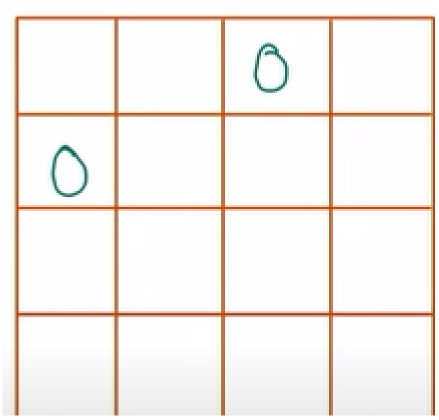
```python
class Solution:
    def setZeroes(self, matrix: List[List[int]]) -> None:
        """
        Do not return anything, modify matrix in-place instead.
        """
        if not matrix:
            return []

        m = len(matrix)
        n = len(matrix[0])

        #we will be taking two arrays to store the info of row and col
        row_z = [False]*m
        col_z = [False]*n

        for row in range(m):
            for col in range(n):
                if matrix[row][col] == 0:
                    row_z[row] = True
                    col_z[col] = True

        for row in range(m):
            for col in range(n):
                if row_z[row] or col_z[col]:
                    matrix[row][col] = 0
```

8. Now we will even optimize space
9. Here we will use first row and first col of matrix ..to store the [T,F] data
10. 

0	1	2	0
3	4	0	2
1	3	1	5
8	7	6	9

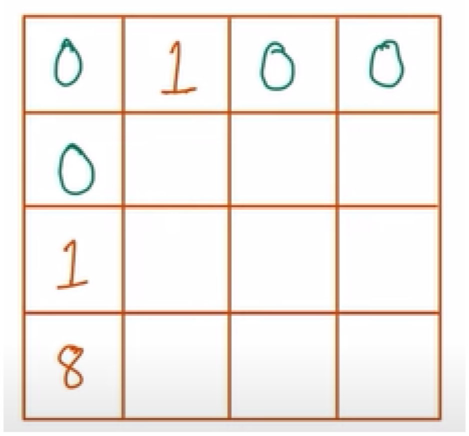


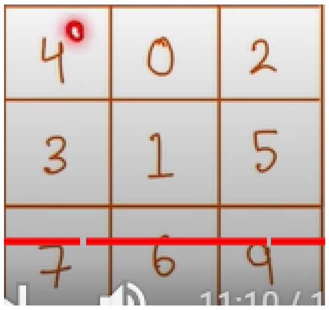
11. Now lets concentrate on this subset of matrix..
12. In this matrix..whenever we encounter ‘0’ then we will make its first_row and first_col as



‘0’

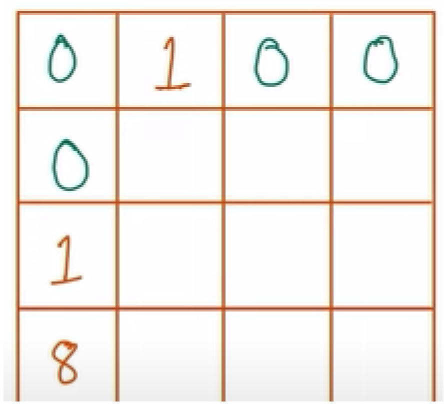
13. Now as there are no more ‘0’ in our subset of matrix
14. Next step would be copying all the ‘0’s from first row and first col





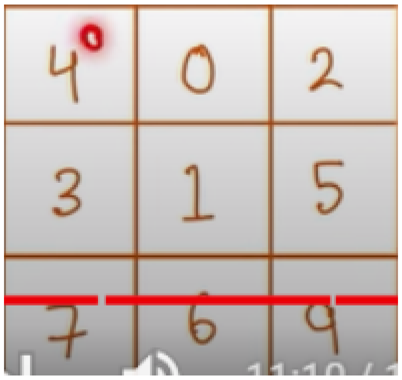
15. Now again we will traverse the submatrix

16. And use our first_row and first col as information matrix

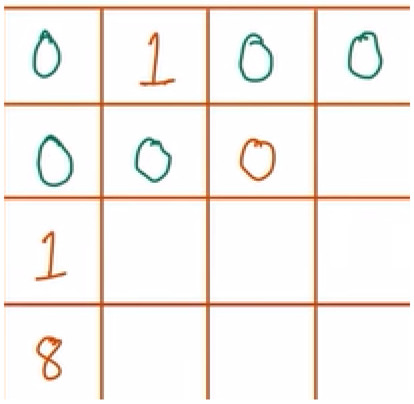


0	1	0	0
0			
1			
8			

17. Now if any of the first col or first row is ‘0’ then we will make the element as ‘0’

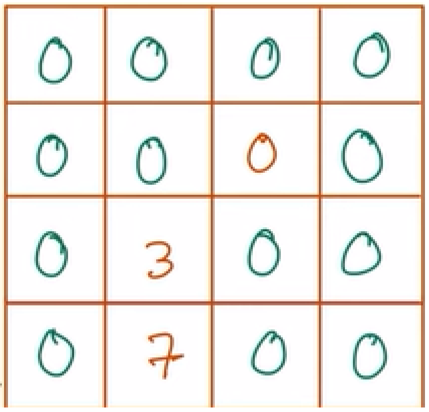


4^{0}	0	2
3	1	5
7	6	9



0	1	0	0
0	0	0	
1			
8			

18. After our iterating our subset ..we’ll gets



0	0	0	0
0	0	0	0
0	3	0	0
0	7	0	0

19. Now we will move to our first_row and first_col