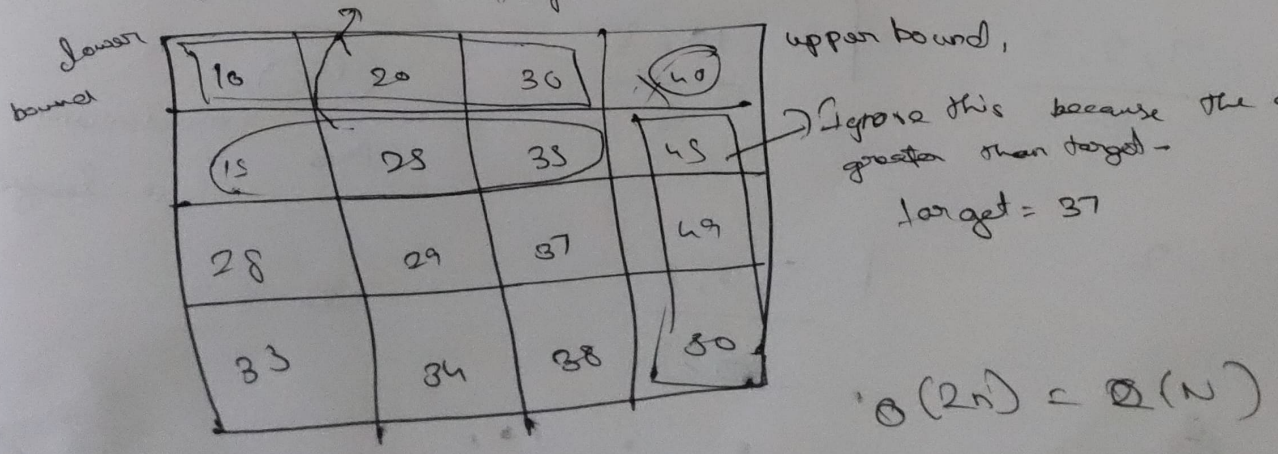
Binary Search in 2D Array:

Searching in Matrices.

Brute force

```
for (r=0 ; r<n ; r++){
    for (c=0; c<n; c++) {
        if (arr[r][c] == target){
        }
    }
}
// It take O(N^2)
```

Q: Matrix is sorted in a rowwise & column wise manner.



$O(2n) \subset O(N)$

Tips:

* In interviews, if they give searching problems try to minimize the search.

Case 1: if element == target //Ans found

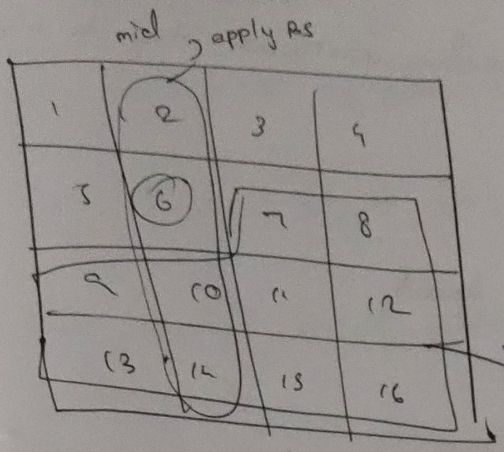
Case 2: if element < target
row++

Case 3: If element > target
col--;

Code

```
static int[] search (int[][] matrix, int target) {
    int row = 0;
    int col = matrix.length - 1;
    while (row < matrix.length && col >= 0)
    {
        if (matrix[row][col] == target) {
            return new int[] {row, col};
        }
        if (matrix[row][col] < target) {
            row++;
        }
        else {
            col--;
        }
    }
    return new int[] {-1, -1};
}
```

Q: Search in a sorted matrix:



Ans:- sorted

case:

① If element == target // Ans

② If element > target

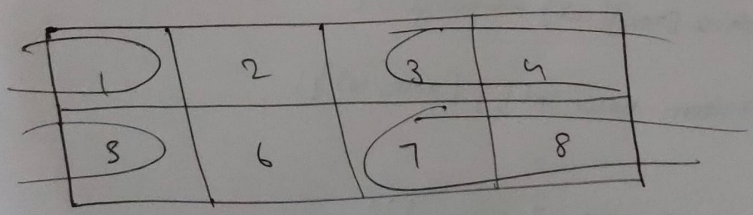
// Ignore rows after it.

③ If element < target

// Ignore above cols.

30:00

In the end 2 rows are remaining:



① check whether the mid row have target.

② consider the four parts if the target does not contain in mid.