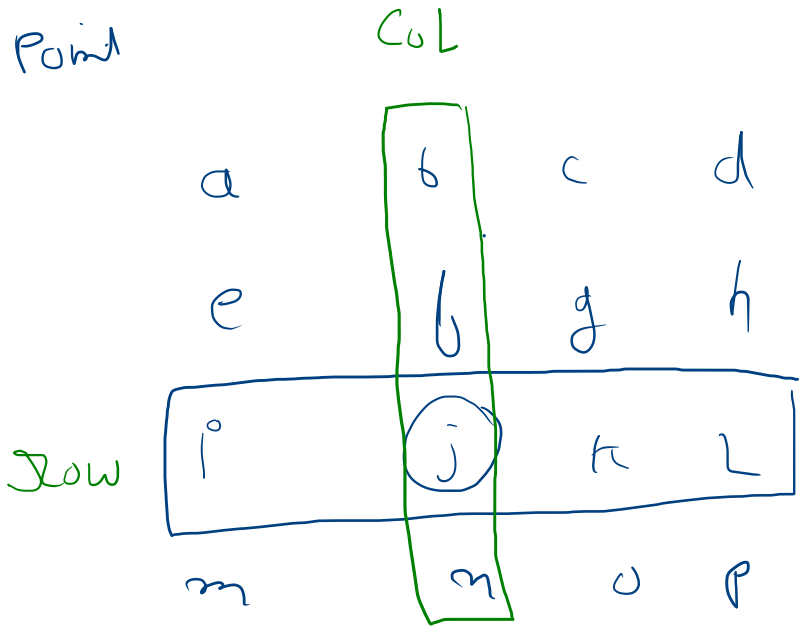
Saddle Price (15 July)

Saddle Point

arr =



least row

max col

b, f, n < j < i, k, L

Saddle Point

Col

arr =

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Saddle

Row

⇒ Low Row
max Col

e → Saddle Pt

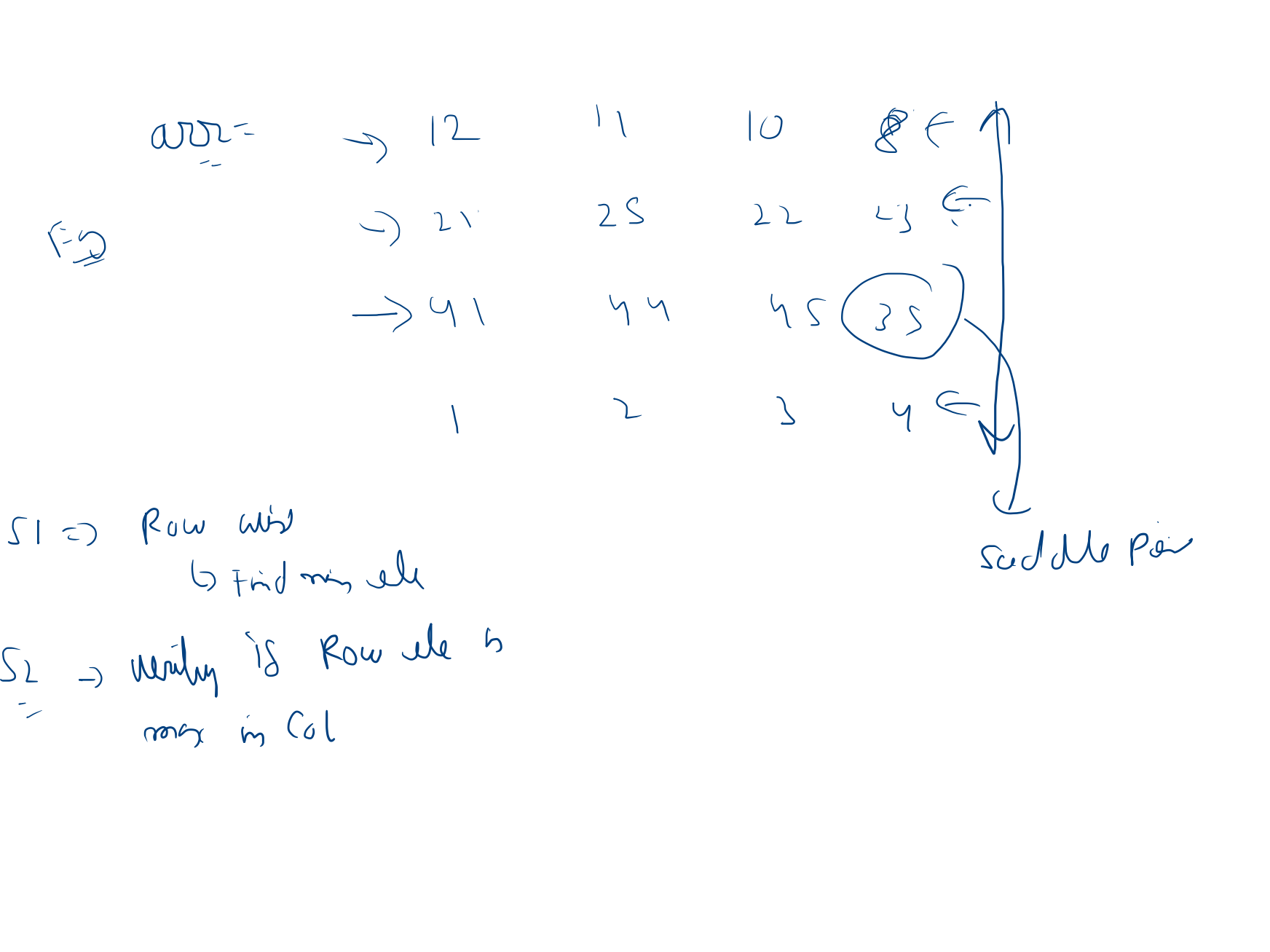
k → Saddle Pt

Statement ⇒ There is only one Saddle Point if it exists.

i < g ← f < e < g

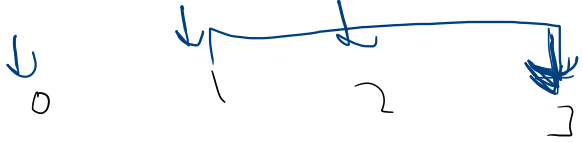
g < i ← g < k < i

one Saddle Point



S1 => Row wise
↳ Find min ele

S2 => Verify is Row ele is max in Col



arr:- 0 → 12 11 10 8

→ 1 → 21 25 22 (23)

→ 2 → 41 44 45 35

3 1 2 3 4

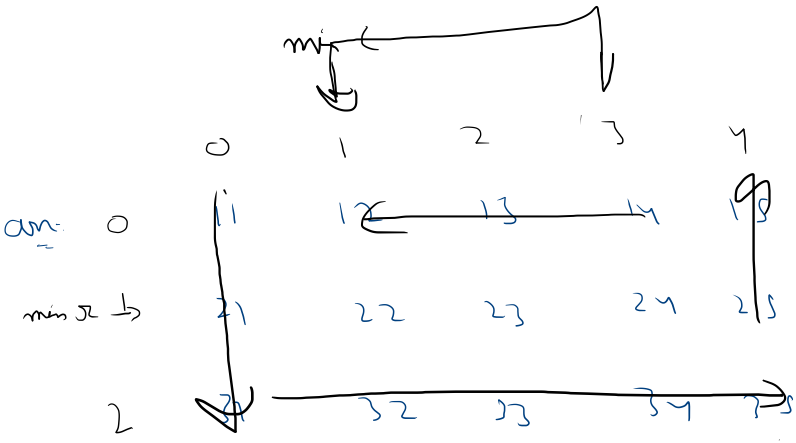
SP = row

min = 21

col = 0

```java
for (int i = 0; i < n; i++) {

    // minEleInRow
    int min = arr[i][0];
    int col = 0;
    for (int j = 1; j < n; j++) {
        if (arr[i][j] < min) {
            min = arr[i][j];
            col = j;
        }
    }

    // VerifyInCol
    boolean isSaddlePoint = true;
    for (int row = 0; row < n; row++) {
        if (min < arr[row][col]) {
            isSaddlePoint = false;
            break;
        }
    }

    if (isSaddlePoint) {
        System.out.println("Saddle point: " + min);
        return;
    }

}
```

Spiral Display (16 july)



minR = 0 | maxR = n-1
minC = 0 | maxC = m-1

11 12 31
32 33 34 35
25 15
14 13 R2

S1 ⇒ for(minR to maxR)
minC cout
minC++

S2 → for(minC to maxC)
maxR cout
maxR--

S3 for(maxR to minR)
maxC cout
maxC--

S4 for(maxC to minC)
minR cout
minR--;