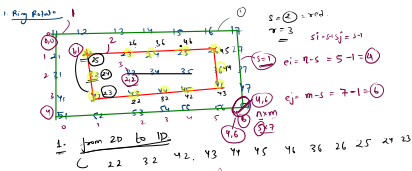
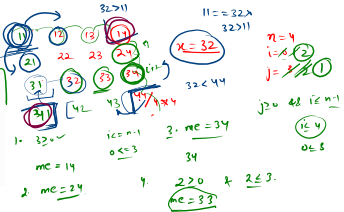


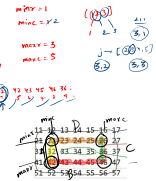
0 to 2  
Search in 2D array  
Smallest Peak



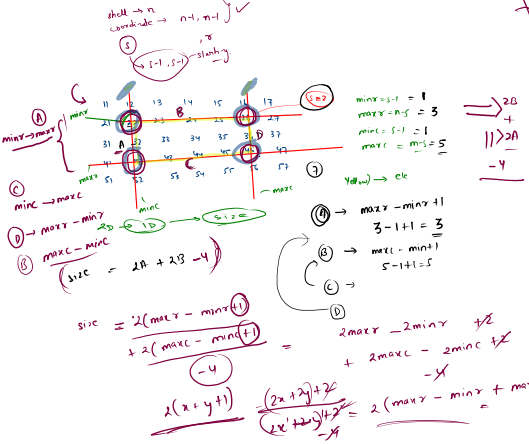
```
public static void convert(int[] arr, int x) {
    //top - right
    int n = arr.length;
    int i = 0;
    while (i < n-1 && arr[i] < arr[i+1]) {
        //convert to int
        arr[i] = (int) Math.pow(2, i);
        i++;
    }
    //bottom - left
    int j = n-1;
    while (j > 0 && arr[j] > arr[j-1]) {
        //convert to int
        arr[j] = (int) Math.pow(2, j);
        j--;
    }
    //print the array
    System.out.println("arr: " + arr);
}
```



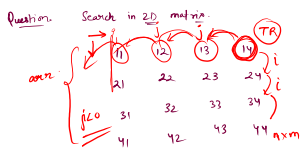
```
int min = Integer.MAX_VALUE;
int max = Integer.MIN_VALUE;
for (int i = 0; i < arr.length; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        if (arr[i][j] < min) {
            min = arr[i][j];
        }
        if (arr[i][j] > max) {
            max = arr[i][j];
        }
    }
}
```



$(max - min + 1)$   
 $(3) - 1 + 1 = 3$   
 $c - a + 2 * (max - min + 1)$   
 $b - b + 2$   
 $n - 5$   
 $5 - 2 = 3$   
 $s - 1$   
 $x = max - min + 1$   
 $y = max - min + 1$   
 $y = 2 * (max - min) + max - min$



break → mainly loop exit  
 continue → 1 iteration skip  
 return → function exit



$2 = 3/3$   
 $41$   
 $36$   
 $2$   
 Approach