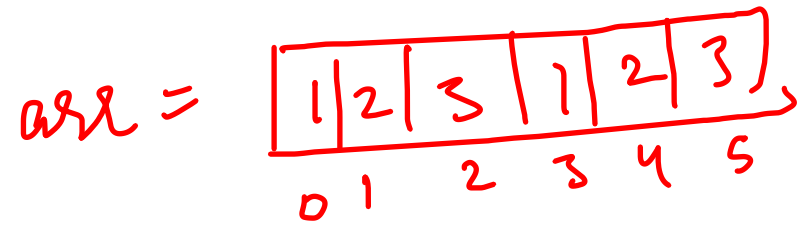
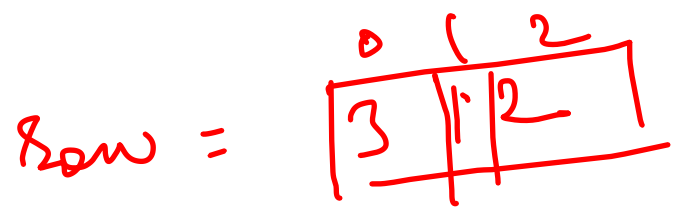
```java
// Function to check if 'row' is a subarray of 'arr'
public static boolean isSubarray(int[] arr, int[] row, int n) {
    for (int i = 0; i <= arr.length - n; i++) {
        boolean match = true;
        for (int j = 0; j < n; j++) {
            if (arr[i + j] != row[j]) {
                match = false;
                break;
            }
        }
        if (match) return true;
    }
    return false;
}
```





i=0 < 3

m-t j=0<3

t arr[0+0] != 3

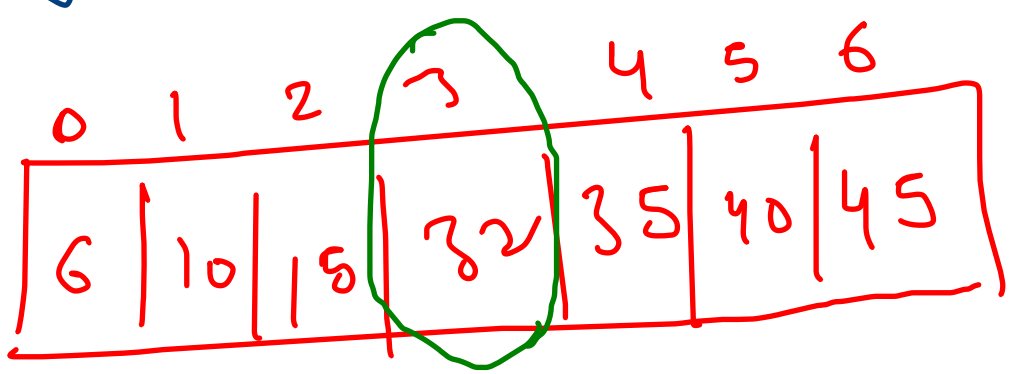
SC → O(n)

TC → O(n²)

```java
        System.out.println(areRowsCircularRotations(mat, n));
    }

    public static String areRowsCircularRotations(int[][] mat, int n) {
        // Create the "rotational space" for the first row
        int[] firstRow = new int[2 * n];
        for (int i = 0; i < n; i++) {
            firstRow[i] = mat[0][i];
            firstRow[i + n] = mat[0][i]; // Duplicate the first row
        }

        // Check each subsequent row
        for (int i = 1; i < n; i++) {
            if (!isSubarray(firstRow, mat[i], n)) {
                return "NO";
            }
        }
        return "YES";
    }
```

Binary Search.

→ array must be sorted



$n = 7$.

$SI = 0$

$EI = 6$

$$mid = \frac{SI + EI}{2} = \frac{0+6}{2} = 3$$

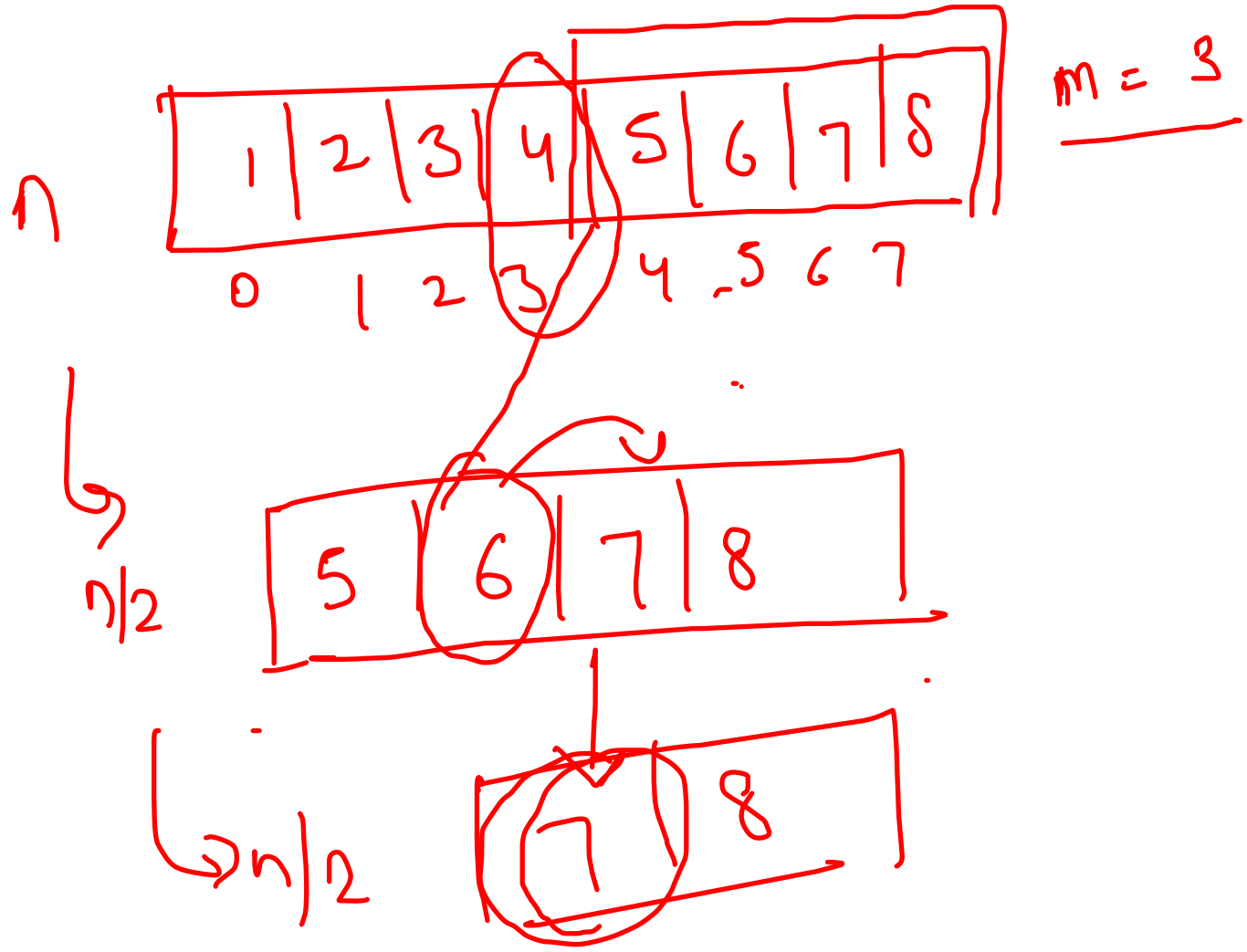
10 < 32 → left

40 > 32 → right

return mid;

l--

r++



TC

best - O(1)

Avg - O(log n)

Worst - O(log n)

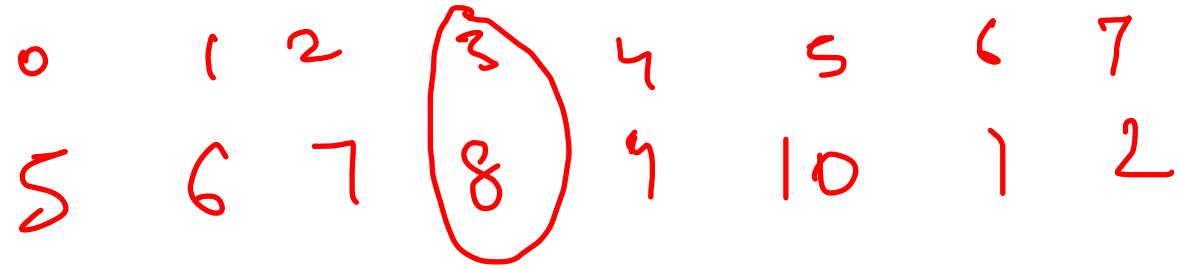
SC

O(1)

```java
public static int rotatedarray(int arr[], int key){
    int left = 0;
    int right = arr.length - 1;

    while(left <= right){
        int mid = (l+r)/2;

        if(arr[mid] == key){
            return mid;
        }

        // which part is sorted
        if(arr[left] <= arr[mid]){
            // left half sorted
            if(key >= arr[left] && key < arr[mid]){
                right = mid - 1;
            }
            else{
                left = mid + 1;
            }
        }
        else{
            // right half is sorted
            if(key > arr[mid] && key <= arr[right]){
                left = mid + 1;
            }
            else{
                right = mid - 1;
            }
        }
    }
    // if key not found
    return -1;
}
```

$TC \rightarrow O(\log n)$

$SC \rightarrow O(1)$



5 < 8 ?