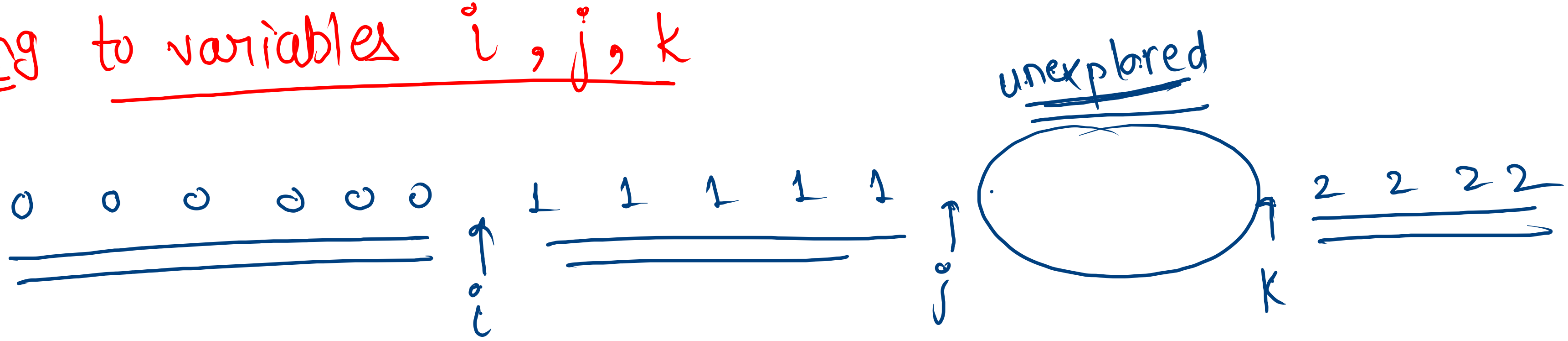
Ques sort012

T.C = O(n)
S.C = O(1)

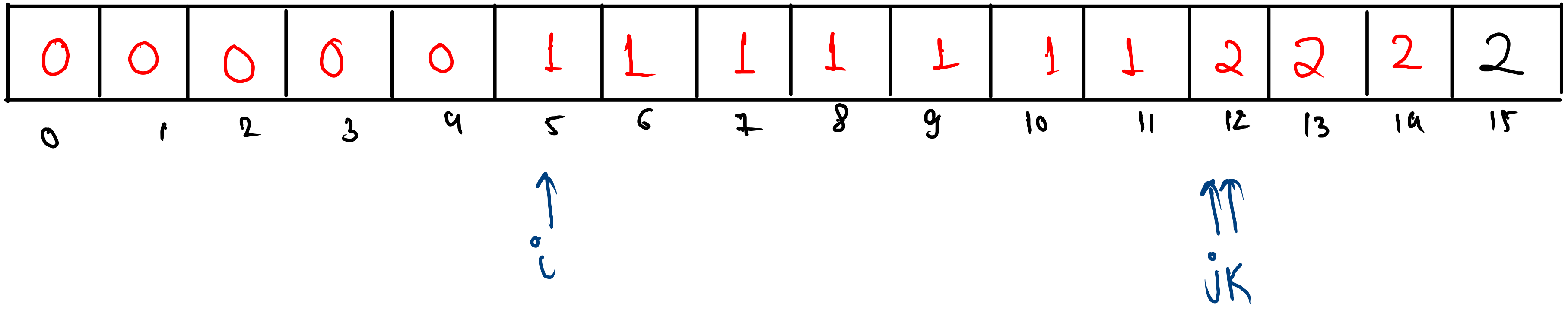
0	1	2	1	0	0	1	2	1	0	1	1	2	0	1	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

meaning to variables i, j, k



rearrange numbers like this

0 0 0 0 0 1 1 1 1 1 1 1 1 2 2 2



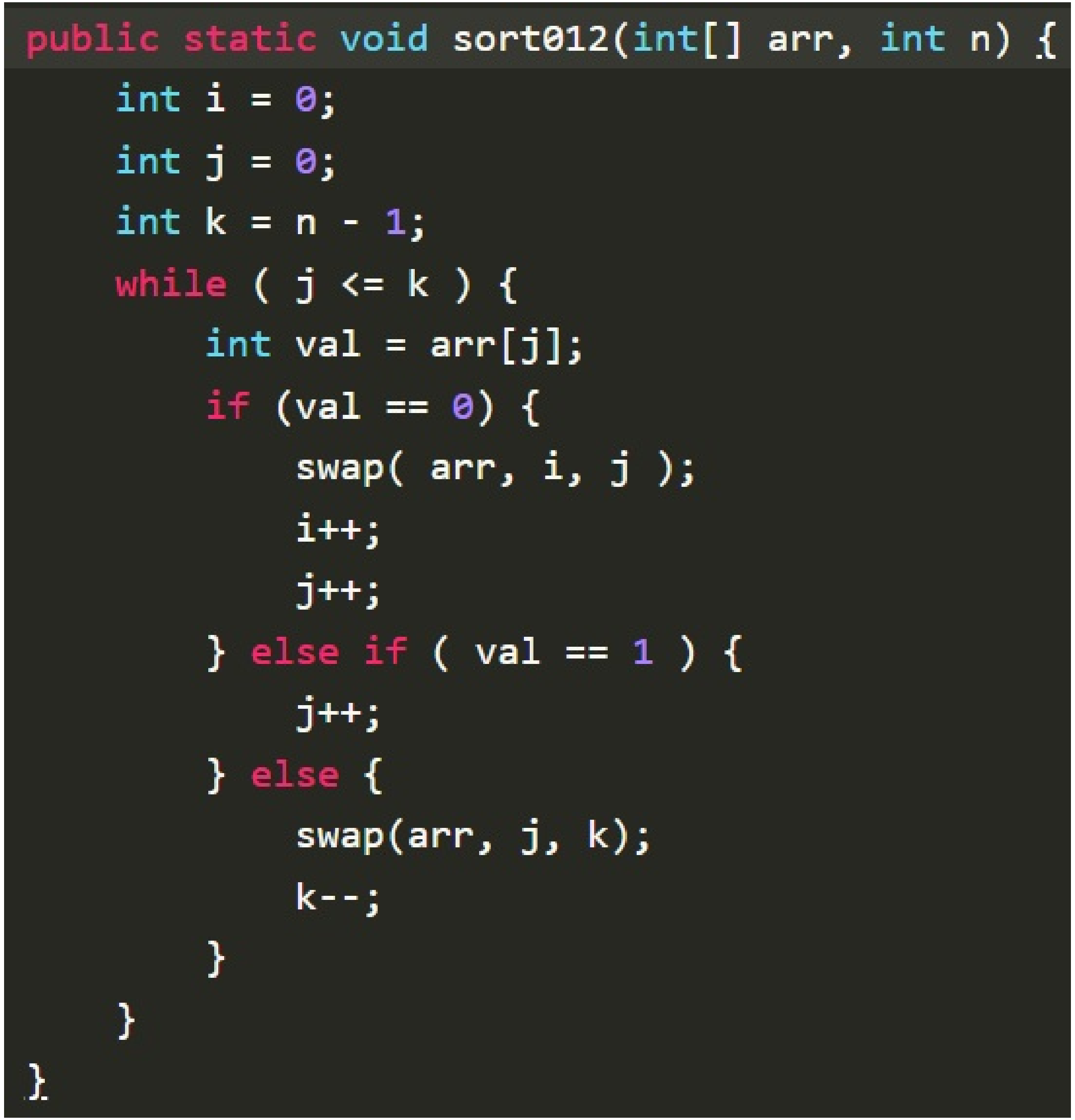
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	1	1	1	1	1	2	2	2	2

psuda code

```
while( j <= k) {
        if ( arr[j] == 0) {
            swap(j, i)
            i++ , j++
        } else if ( arr[j] == 1){
            j++;
        } else {
            swap(j, k);
            k--;
        }
}
```

$\boxed{j{+}{+}\,\alpha}$ ~~αα~~ ∝α∝

code



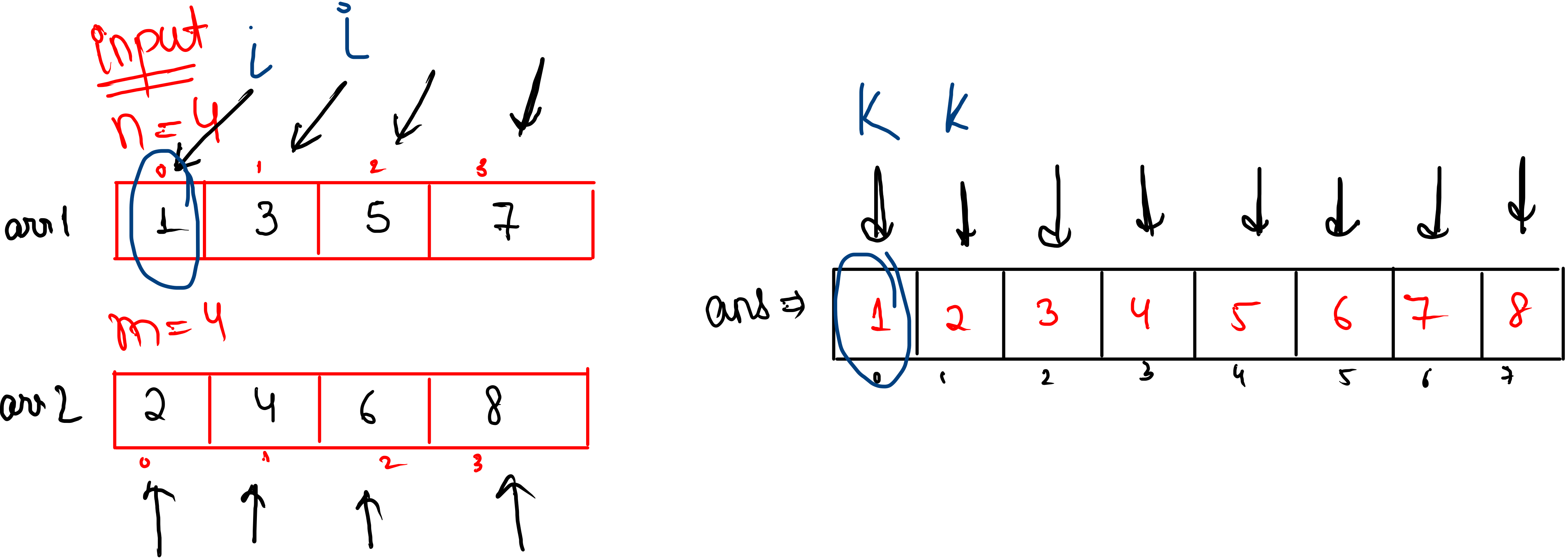
```
public static void sort012(int[] arr, int n) {
    int i = 0;
    int j = 0;
    int k = n - 1;
    while ( j <= k ) {
        int val = arr[j];
        if (val == 0) {
            swap( arr, i, j );
            i++;
            j++;
        } else if ( val == 1 ) {
            j++;
        } else {
            swap(arr, j, k);
            k--;
        }
    }
}
```

HW_Merge two sorted arrays

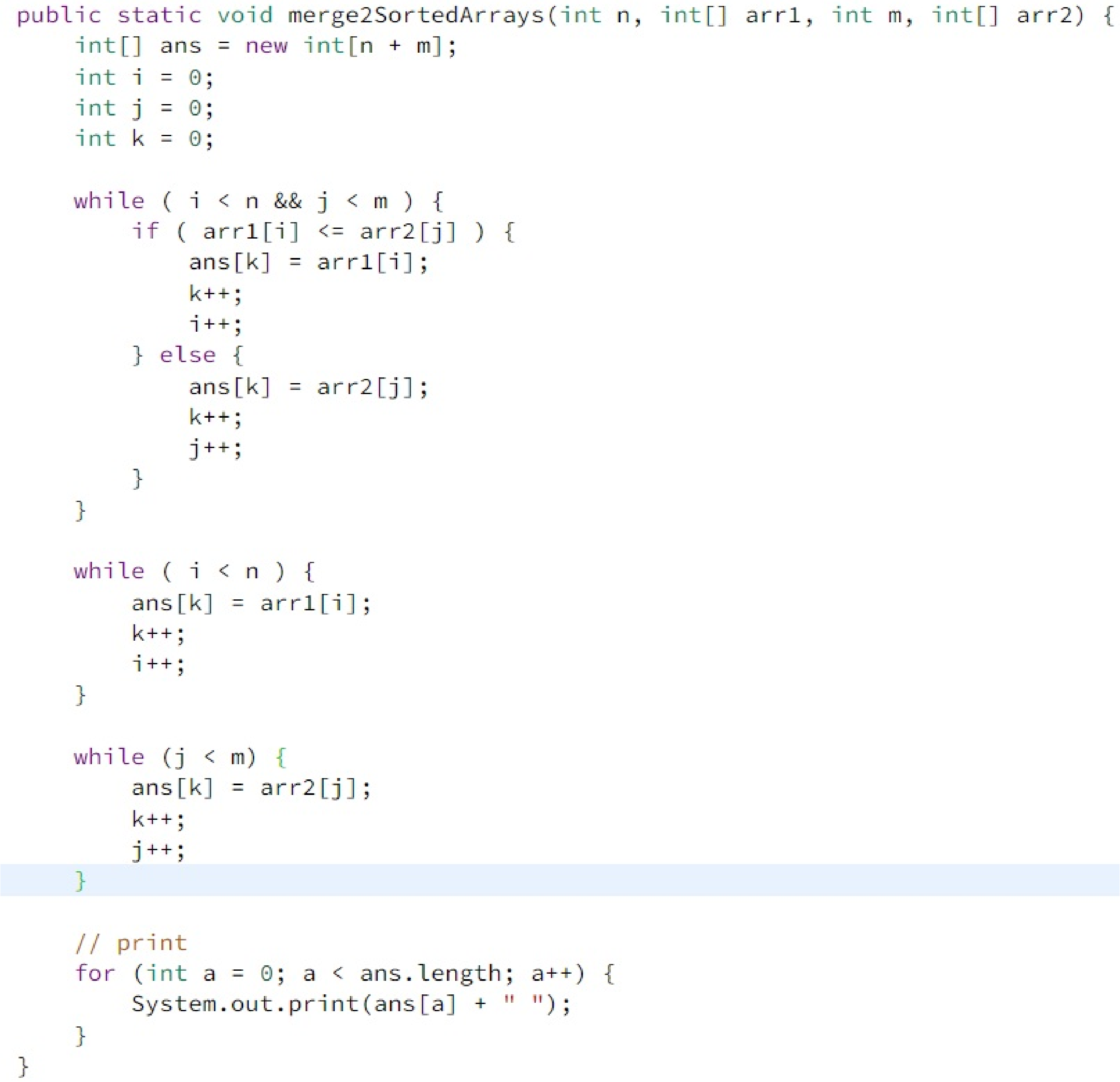
Note:-

increasing order :- 1 2 3 5 6 7 9

non decreasing :- 1 2 2 5 5 5 6



Code



```java
public static void merge2SortedArrays(int n, int[] arr1, int m, int[] arr2) {
    int[] ans = new int[n + m];
    int i = 0;
    int j = 0;
    int k = 0;

    while ( i < n && j < m ) {
        if ( arr1[i] <= arr2[j] ) {
            ans[k] = arr1[i];
            k++;
            i++;
        } else {
            ans[k] = arr2[j];
            k++;
            j++;
        }
    }

    while ( i < n ) {
        ans[k] = arr1[i];
        k++;
        i++;
    }

    while (j < m) {
        ans[k] = arr2[j];
        k++;
        j++;
    }

    // print
    for (int a = 0; a < ans.length; a++) {
        System.out.print(ans[a] + " ");
    }
}
```