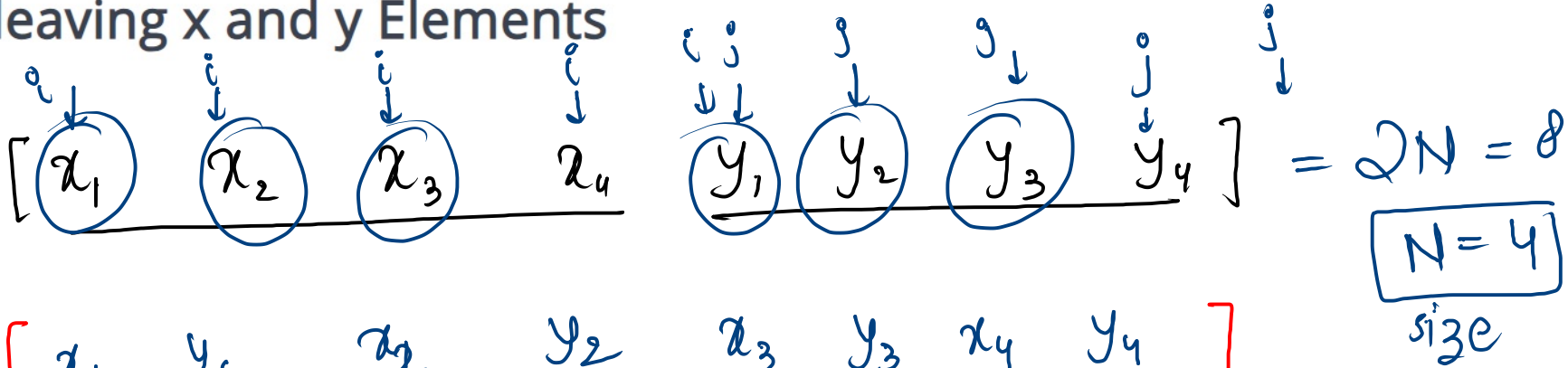
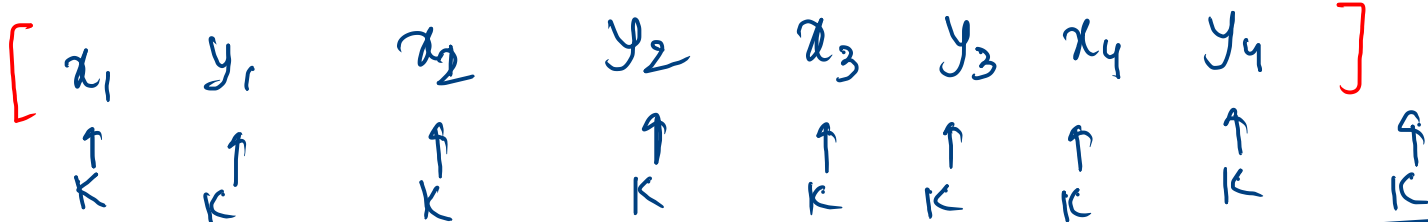


# Interleaving x and y Elements

arr



ans



$$2 * \text{size} = 2N$$

```
public static int[] interleaving(int[] arr, int size) {
    int i = 0;
    int j = size;
    int[] ans = new int[2 * size];
    int k = 0;
    while (k < ans.length) {
        ans[k] = arr[i];
        k++;
        i++;

        ans[k] = arr[j];
        k++;
        j++;
    }
    return ans;
}
```

iterations = 4  
(N)

2N

# Zeros and Ones (inplace)

arr

0	1	2	3	4	5	6	7	8	9
1	0	1	0	0	1	1	0	1	1

↑ ↑  
i j

0 (i) 1 (j) unexplored

Faith) i = all elements before i will always be 0

j = all element before j and after i will always be 1

all other = unexplored

arr

0	0	0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

arr

0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	1	1	1	1	1

↑  
i
↑  
j

i = 0, j = 0

a [ if element at j pointer is 1  
j++;

b [ else if element at j pointer is 0  
swap(i, j)  
i++;  
j++;

```

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    int[] ans = swap01(arr, n);

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

```

```

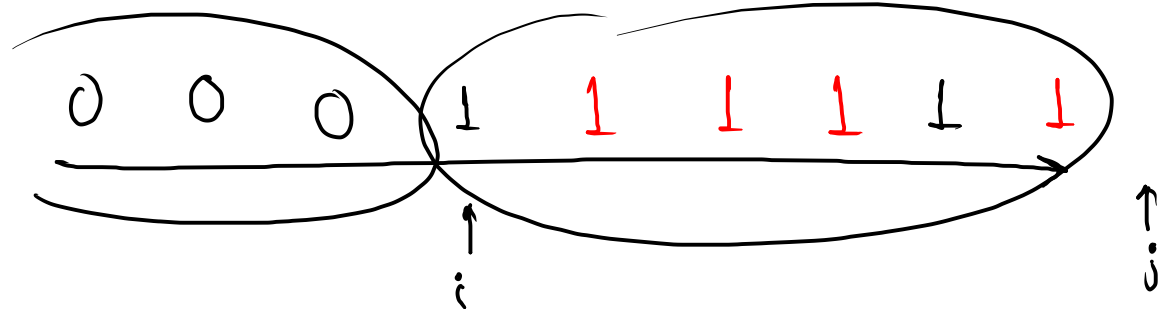
public static int[] swap01(int[] arr, int n) {
    int i = 0;
    int j = 0;
    while (j < n) {
        a[ if (arr[j] == 1) {
            j++;
        } else {
            b[ swap(arr, i, j);
               i++;
               j++;
            }
        }
    }
    return arr;
}

```

```

public static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

```



$$T.C = \underline{\underline{O(N)}}, S.C = \underline{\underline{O(1)}}$$

Note:-

1 1 1 1 1 0 0 0 0

even

odd

odd

even

O(1)

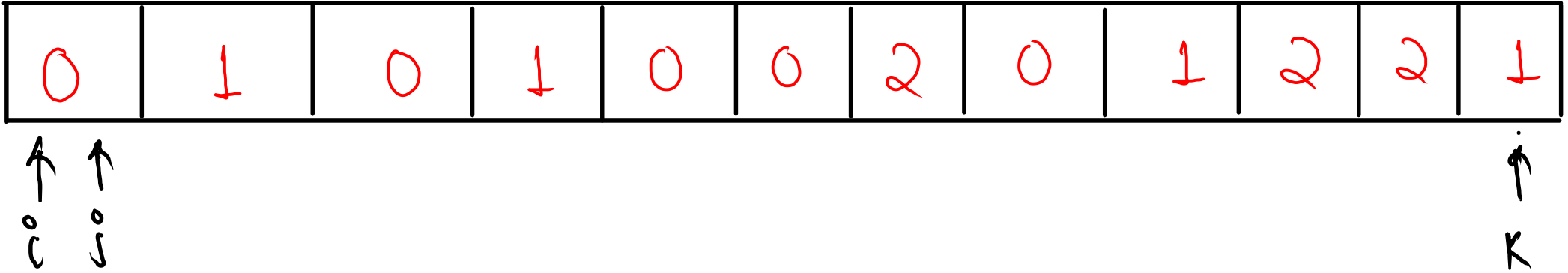
# Sort 0 1 2

0	1	0	1	0	0	2	0	1	2	2	1
---	---	---	---	---	---	---	---	---	---	---	---



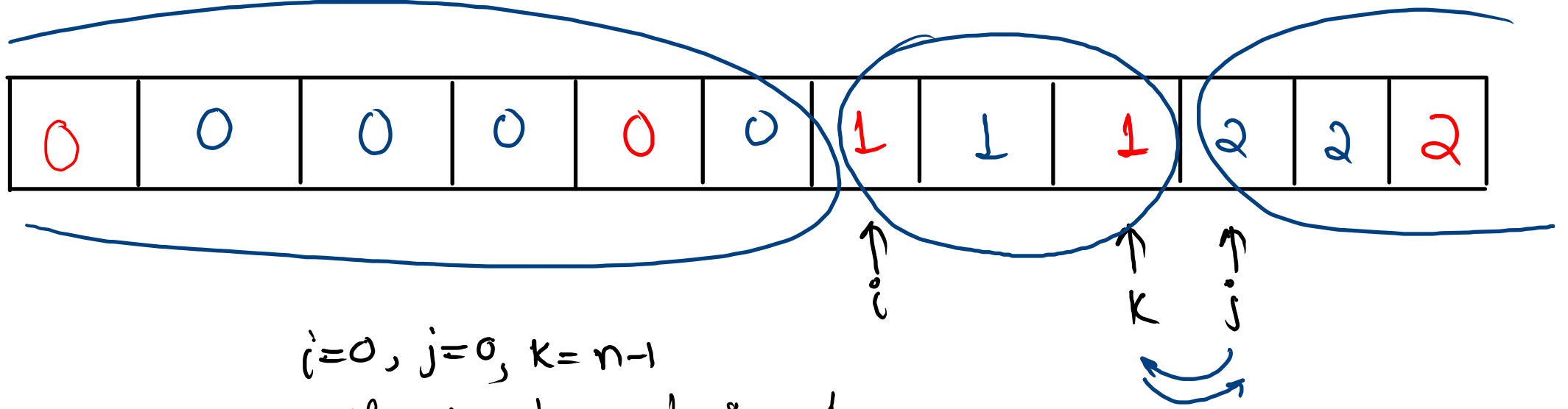
partition → 0  
1  
2  
unexplored

where exactly our i, j, and  
k will start from  
partition



0      (i)      1      (j)      unexpl.      (k)      2

faith:-  $i$  = all elements before  $i$  will be 0  
 $j$  = all elements before  $j$  and after  $i$  will be 1  
 $k$  = all elements after  $k$  will be 2  
 remaining = unexplored



$i=0, j=0, k=n-1$

pseudo code

```

a [ if j element is 1
    j++
    else if j element is 0
b [ swap(i, j)
    i++
    j++
c [ else if j element is 2
    swap(j, k)
    k--j
  
```

stop when both  
j and k pointer  
crossed each  
other

```
public static int[] swap012(int[] arr, int n) {  
    int i = 0;  
    int j = 0;  
    int k = n - 1;  
    while (j <= k) {  
        if (arr[j] == 1) {  
            j++;  
        } else if (arr[j] == 0) {  
            swap(arr, i, j);  
            i++;  
            j++;  
        } else {  
            swap(arr, j, k);  
            k--;  
        }  
    }  
    return arr;  
}
```

$j > k \rightarrow \text{stop}$



## Rotate Right

1 2 3 4 5 6 7

$K=2$

$0 \leq K \leq \underline{1000000}$

$K=1$  7 1 2 3 4 5 6

$K=2$

6 7 1 2 3 4 5

$K=3$

5 6 7 1 2 3 4