Shift right:

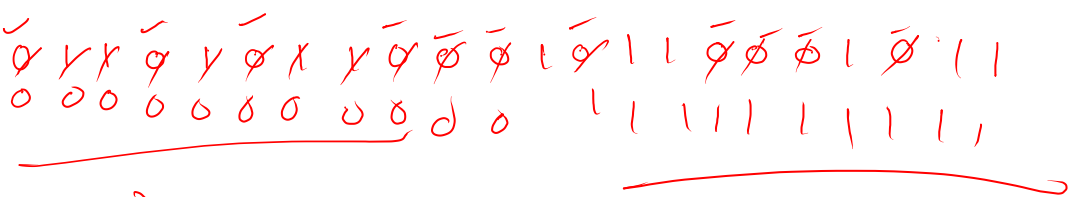
K = 2

k = 0 [1 2 3 4 5

```java
class Solution {
    public void rotate(int[] nums, int k) {
        int n = nums.length;
        k = k % n;
        reverse(nums, 0, n - k - 1);
        reverse(nums, n - k, n - 1);
        reverse(nums, 0, n - 1);
    }

    public void reverse(int[] arr, int s, int e) {
        while(s < e) {
            int temp = arr[s];
            arr[s] = arr[e];
            arr[e] = temp;
            s ++;
            e --;
        }
    }
}
```

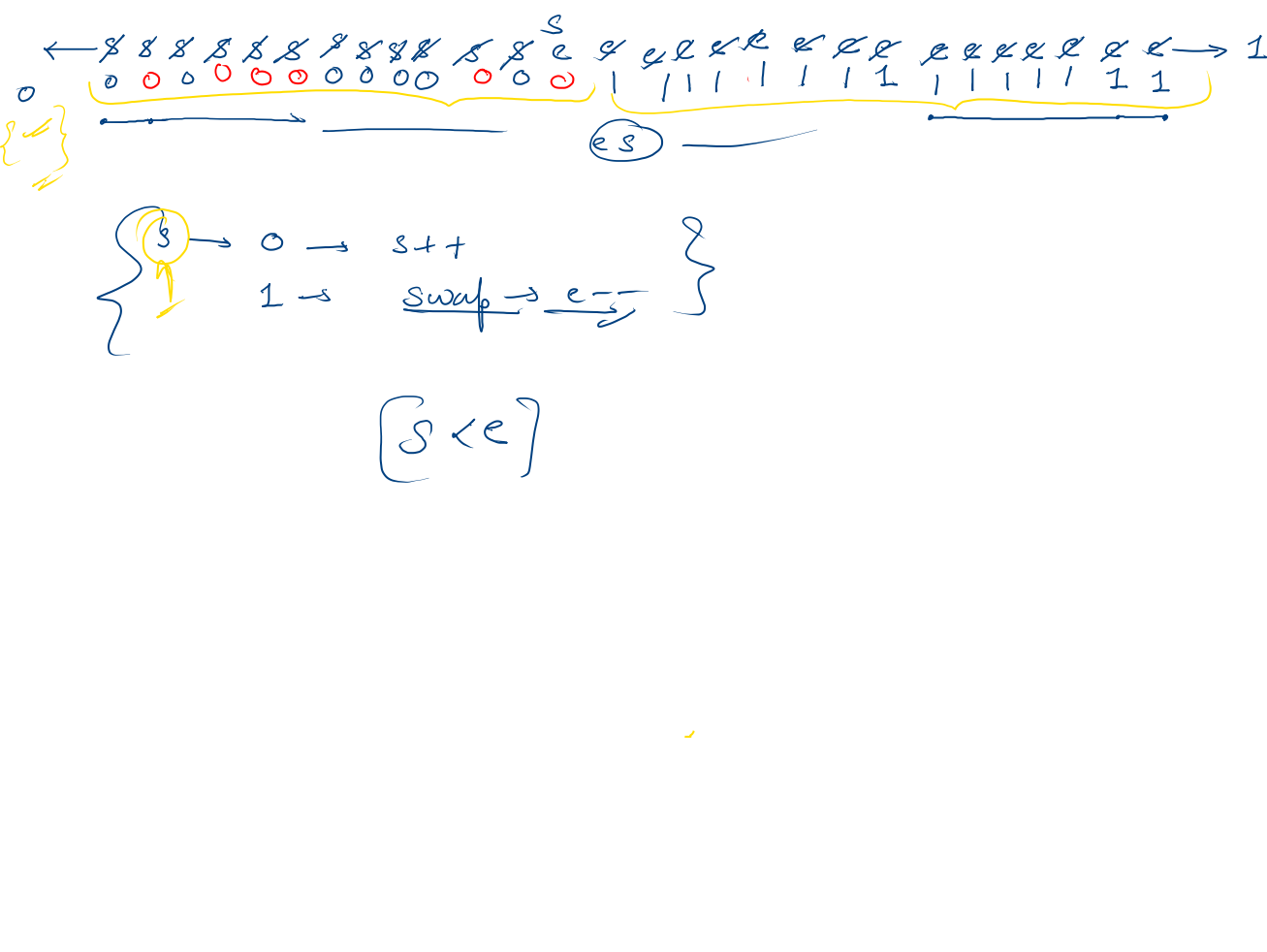
0-1

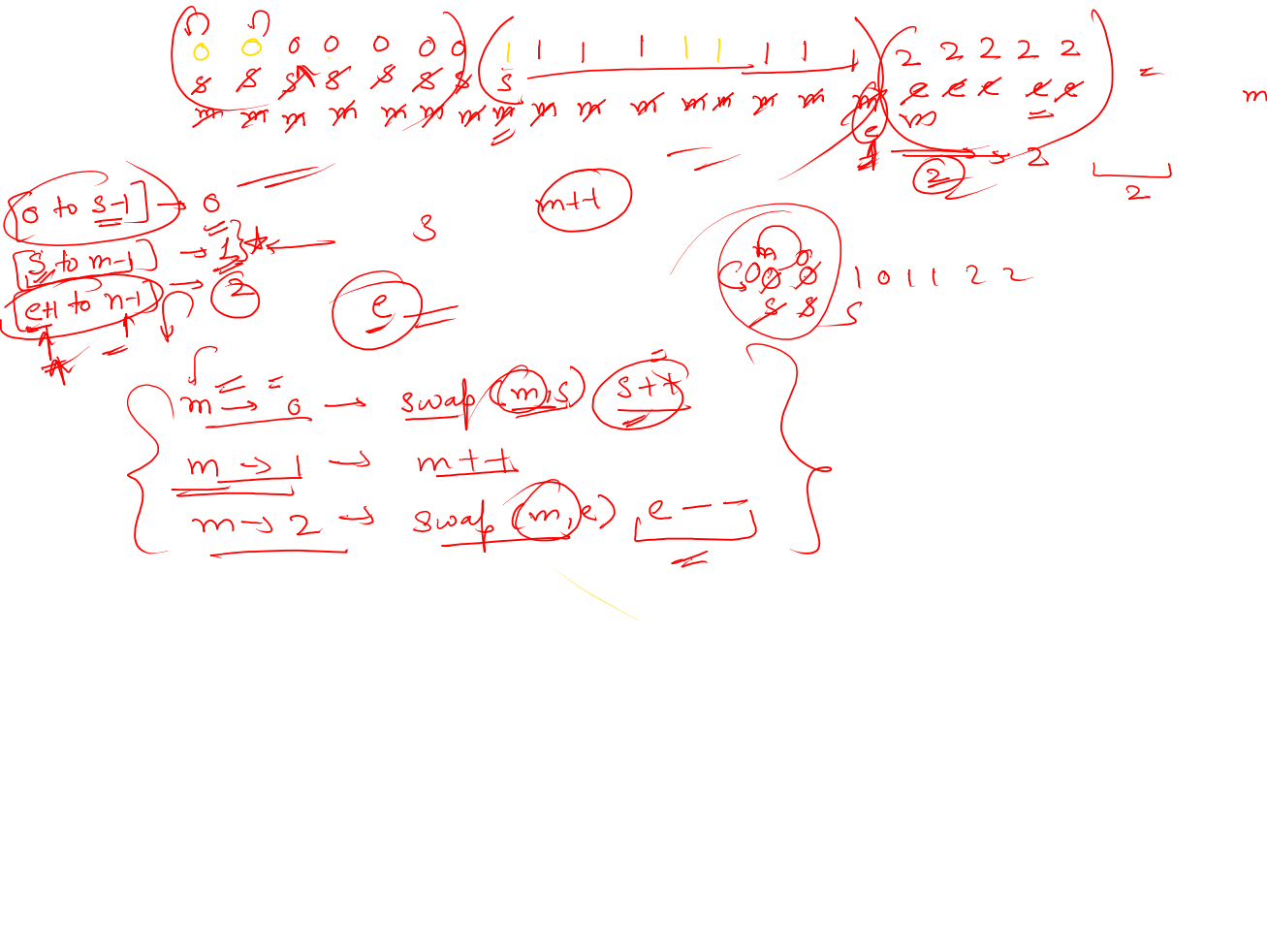


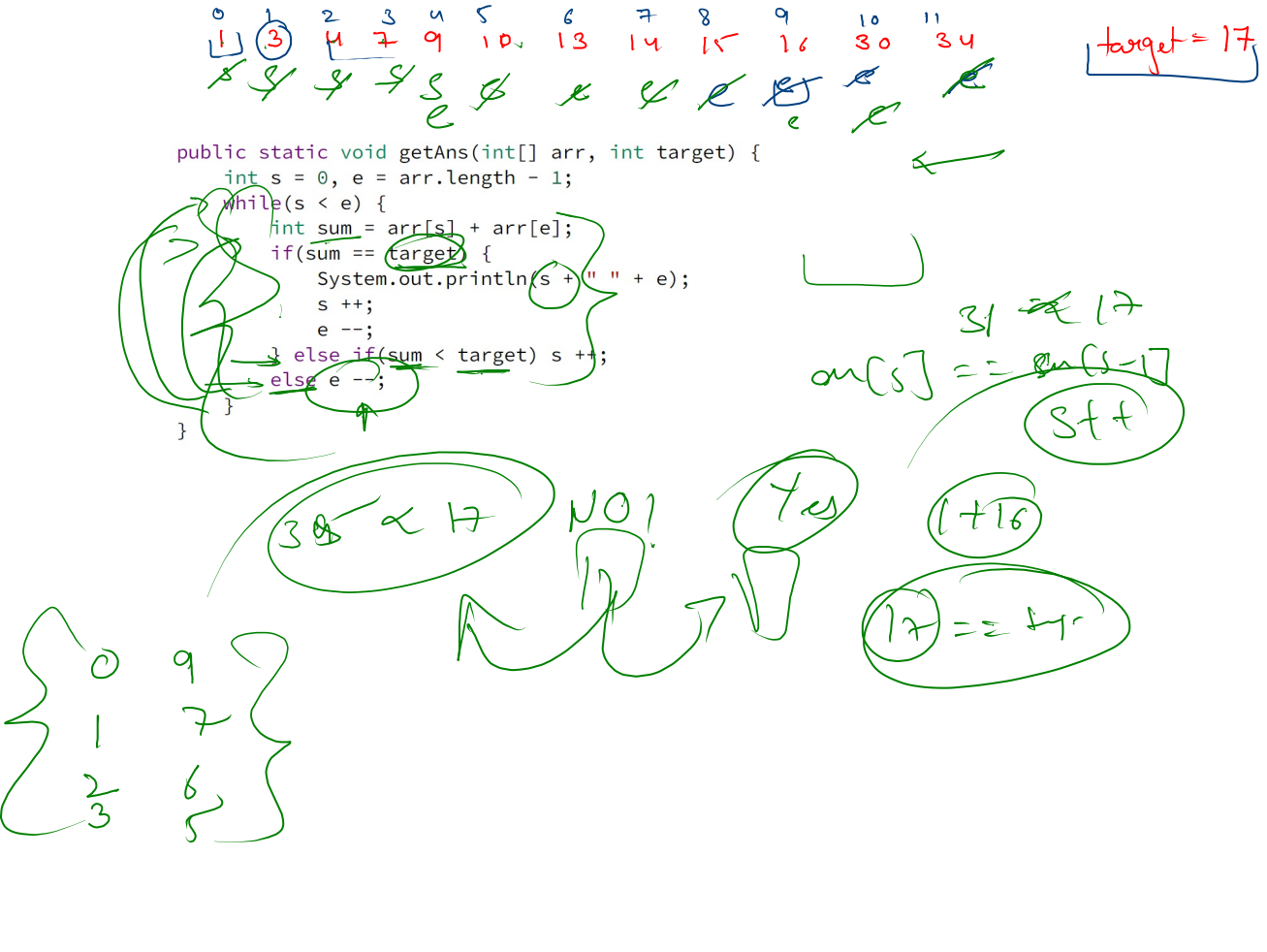
Method 1

O(n)

↳ 2 traversals


s
e
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1
es
s
0 → s++
1 → swap → e--
[s < e]


0 to s-1
s to m-1
e+1 to n-1
m+1
s
e
swap (m, s)
s++
m++
swap (m, e)
e--


0 1 2 3 4 5 6 7 8 9 10 11
1 3 4 7 9 10 13 14 15 16 30 34
target = 17
public static void getAns(int[] arr, int target) {
int s = 0, e = arr.length - 1;
while(s < e) {
int sum = arr[s] + arr[e];
if(sum == target) {
System.out.println(s + " " + e);
s ++;
e --;
} else if(sum < target) s ++;
else e --;
}
}
30 < 17
NO!
Yes
1 + 16
17 == tgt
0 9
1 7
2 6
3 5