Sort 0, 1, 2

DNF Sort

Brute force approach:

Use simple sort algo. or sort built in function

T.C. : $O(N \log N)$

Better approach:

Take count of 0, 1, 2 in 3 variable and then push 0, 1, 2 in the array.

T.C. : $O(N) + O(N)$

Optimal Approach:

Use of DNF sort algorithm:

D → Dutch — Red → 0

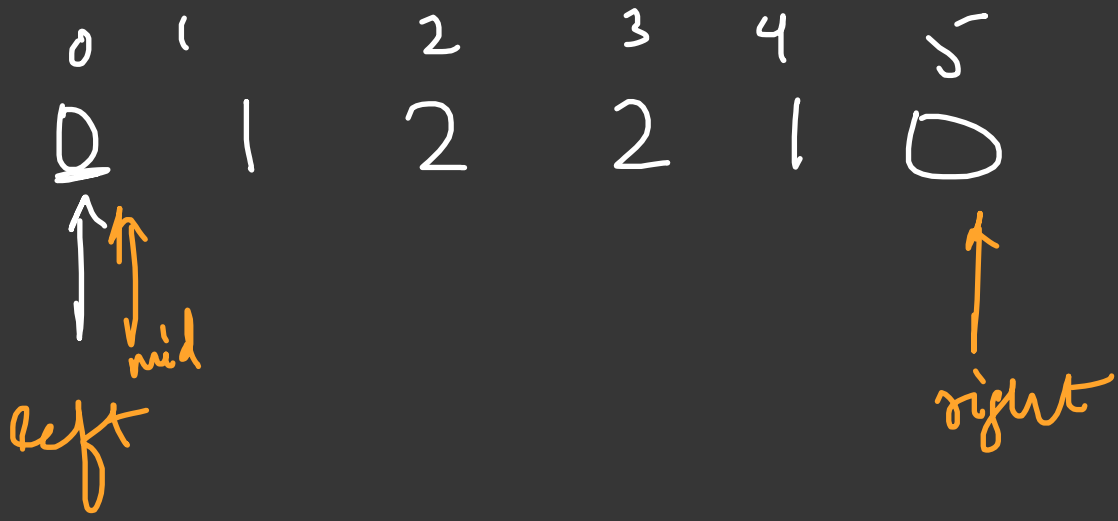
N → Nation — ~~White~~ Blue → 1

F → Flag — ~~White~~ Blue → 2

We maintain 3 pointer

left, mid, right

eg:

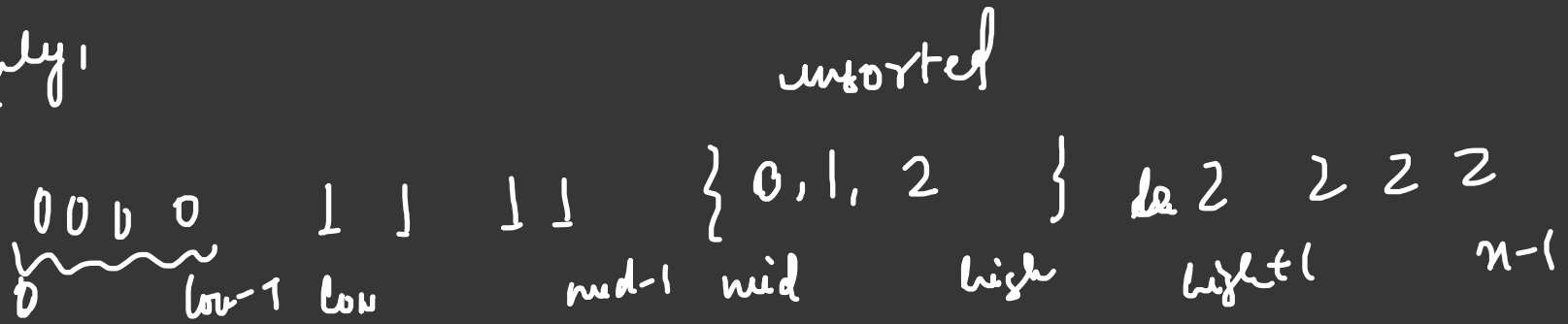


$[0 \ldots\ low-1] = 0$

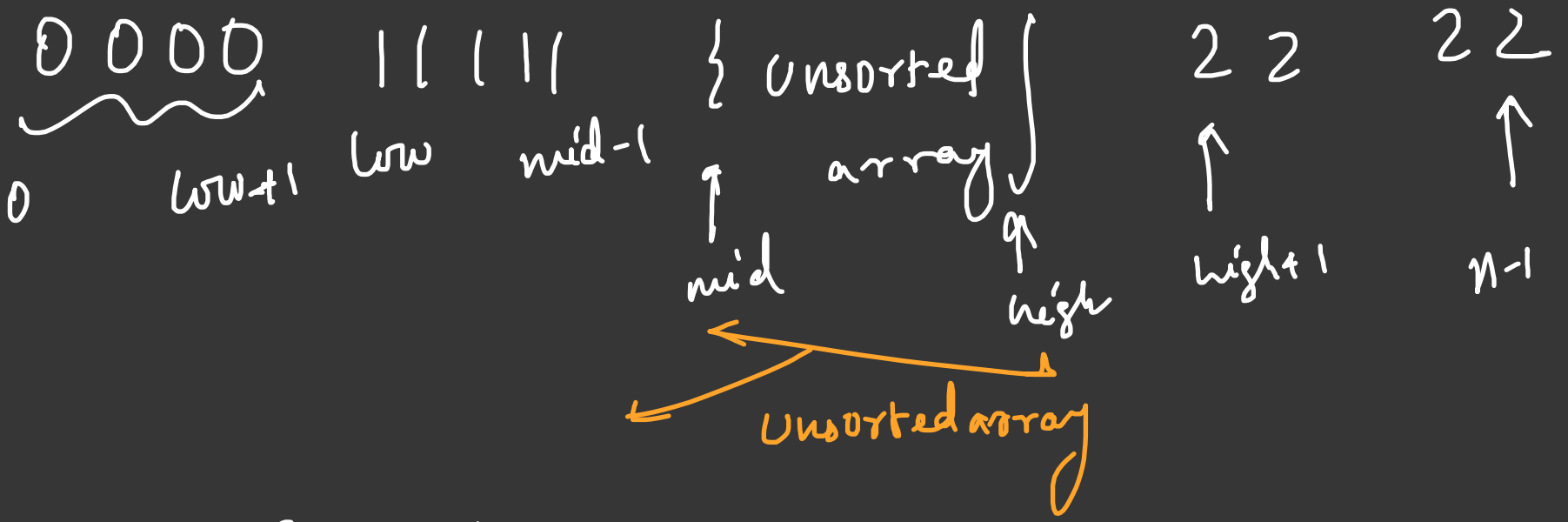
$[low \ldots\ mid-1] = 1$

$[high+1 \ldots\ n-1] = 2$

hypothecially:



Observation:




```
if(arr[mid] == 0)
    swap(left, mid)
    left++, mid++
```

```
if( arr[mid] == 1)
    mid++;
if (arr[mid] == 2)  swap( &right, mid)
                    right++;
```