1) 012 - Sort

Brute force

- Count number of 0's, 1's, 2's in array
- Update the array with respective counts

I/p:

0	2	1	2	0

cnt 0 = 2 , cnt 1 = 1, cnt 2 = 2

↑ make first 2 elements 0

↑ make 3rd element 1

↑ make last elements 2

O/p:

0	0	1	2	2

T.C → O(n)

S.C → O(1)

Optimised Solution (Dutch National flag logic)

- Use 3 ptrs - l, m, h

elems b/w 0 to l-1 ⇒ 0

elems b/w l to m-1 ⇒ 1

elems b/w m to h-1 ⇒ ?

elems b/w h to n-1 ⇒ 2

```
0 0 0  1 1  ? ? ? ?  2 2 2
↑      ↑    ↑        ↑   ↑
0      l    m        h   n-1
```

~ Do operation in window m to h

```
if arr[m] == 0 → swap(arr[m], arr[l]);
                 l++;

if arr[m] == 1 → m++;

if arr[m] == 2 → swap(arr[m], arr[h]);
                 h--;
```

Dry run

0 2 1 2 0

l m h

0 2 1 2 0

l m h

a[m]==2
swap(a[m], a[h)
h--

0 0 1 2 2

l m h

a[m]==0
swap(a[m], a[l]
l++, m++

0 0 1 2 2

l m h

a[m]==1
m++

0 0 1 2 2

l m h

a[m]==2
swap(a[m], a[h]
h--

0 0 1 2 2

l h m

m > h
break out

Optimised code

```
int l=0, h=n-1, m=0;
while (m <= h) {
   switch (a[m]) {
        case 0:
          swap(a[m], a[l]);
          m++; l++;
          break;

        case 1:
          m++;
          break;
        case 2:
          swap(a[m], a[h]);
          h--;
          break;
   }
}
```