Approach

Approach: Using two pointer low and high we denote the positions of zeros and ones. Low denotes the last position where the zero occures and high denotes the last position where one occures.

When we traverse we check that if the arr[low] is '0' then nothing to do it is in correct place so just increment low,

but if it is '1' then swap with arr[high] and decrement position of high as one element is being positioned correctly at high.

0 0 0 0 0 1 1 1 0 0 1 0 1 1

low high

0 0 0 0 0 0 1 1 0 0 1 1 1 1

low high

0 0 0 0 0 0 1 1 0 0 1 1 1 1

low high

0 0 0 0 0 0 0 1 0 1 1 1 1 1

low high

0 0 0 0 0 0 0 0 1 1 1 1 1 1

low/ high [End condition]

Time complexity = O(n)

Space complexity = O(1)