DNF Sort

Problem

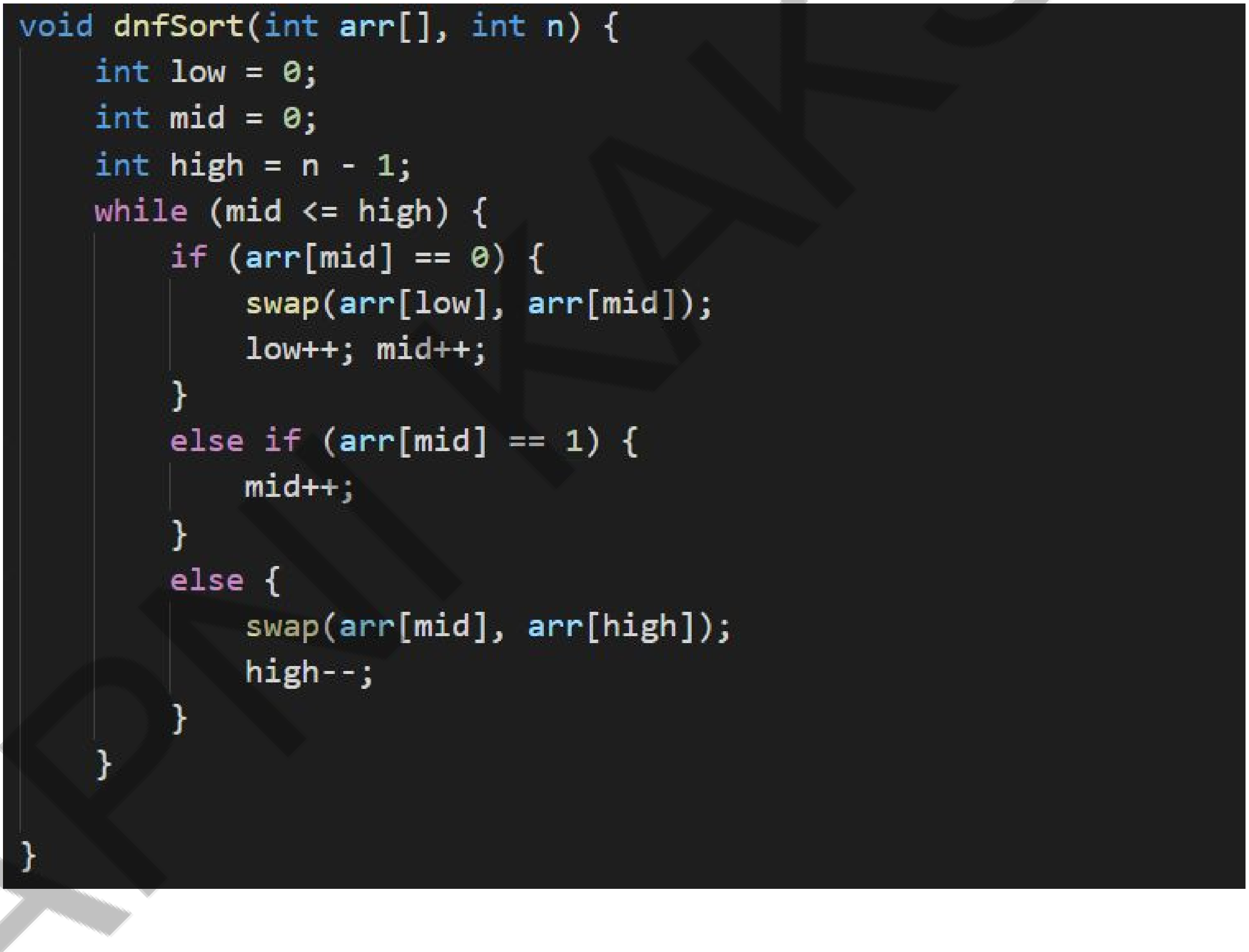
Given an array containing only 0,1 and 2. You have to sort the array in O(N) time, single pass, and O(1) space.

Input: Given an array, containing only 0,1, and 2.

Output: print the sorted array.

Time Complexity: O(N), single pass

Space Complexity: O(1)

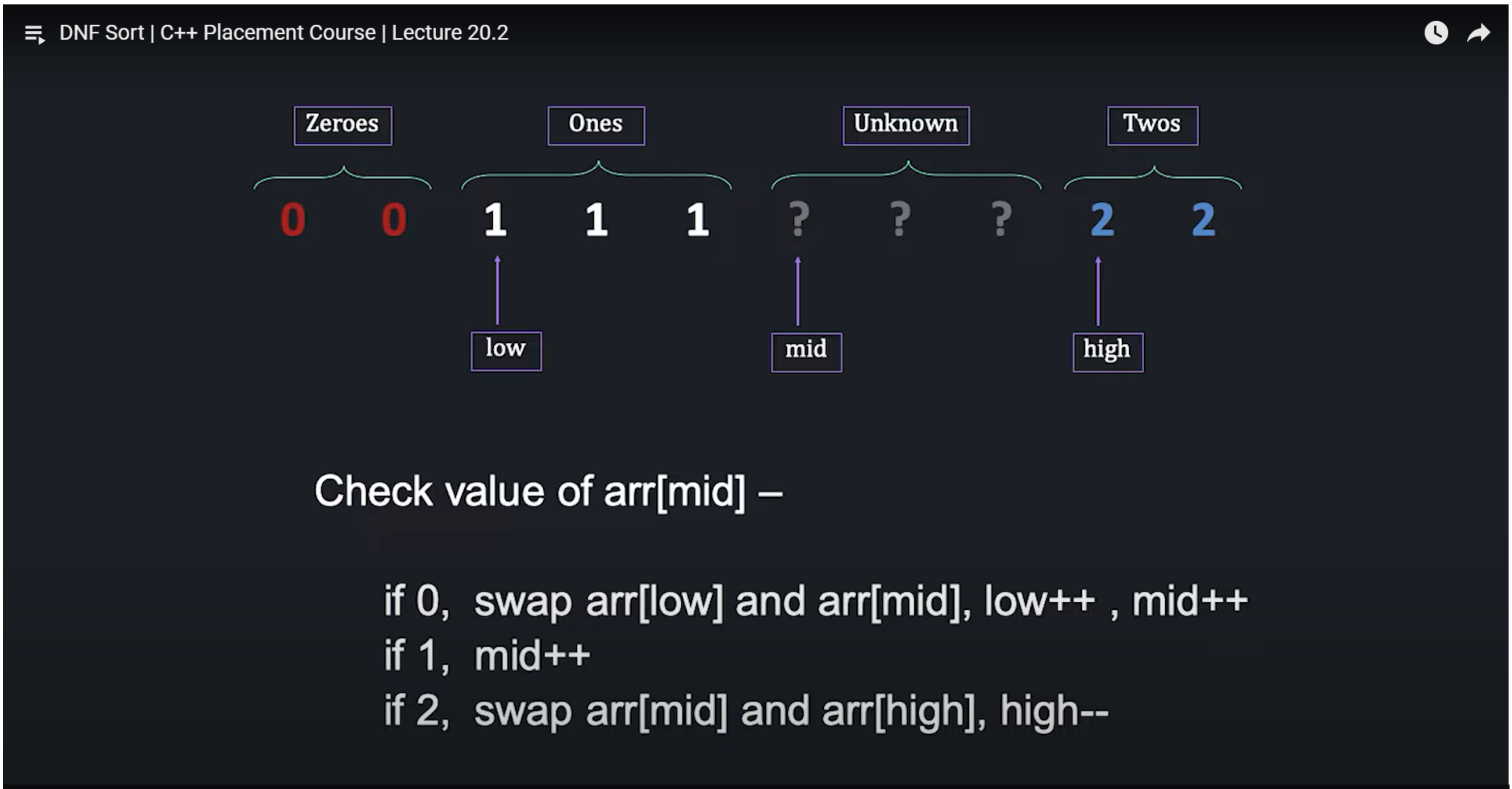


```
void dnfSort(int arr[], int n) {
    int low = 0;
    int mid = 0;
    int high = n - 1;
    while (mid <= high) {
        if (arr[mid] == 0) {
            swap(arr[low], arr[mid]);
            low++; mid++;
        }
        else if (arr[mid] == 1) {
            mid++;
        }
        else {
            swap(arr[mid], arr[high]);
            high--;
        }
    }

}
```

Online Judge for the above problem:

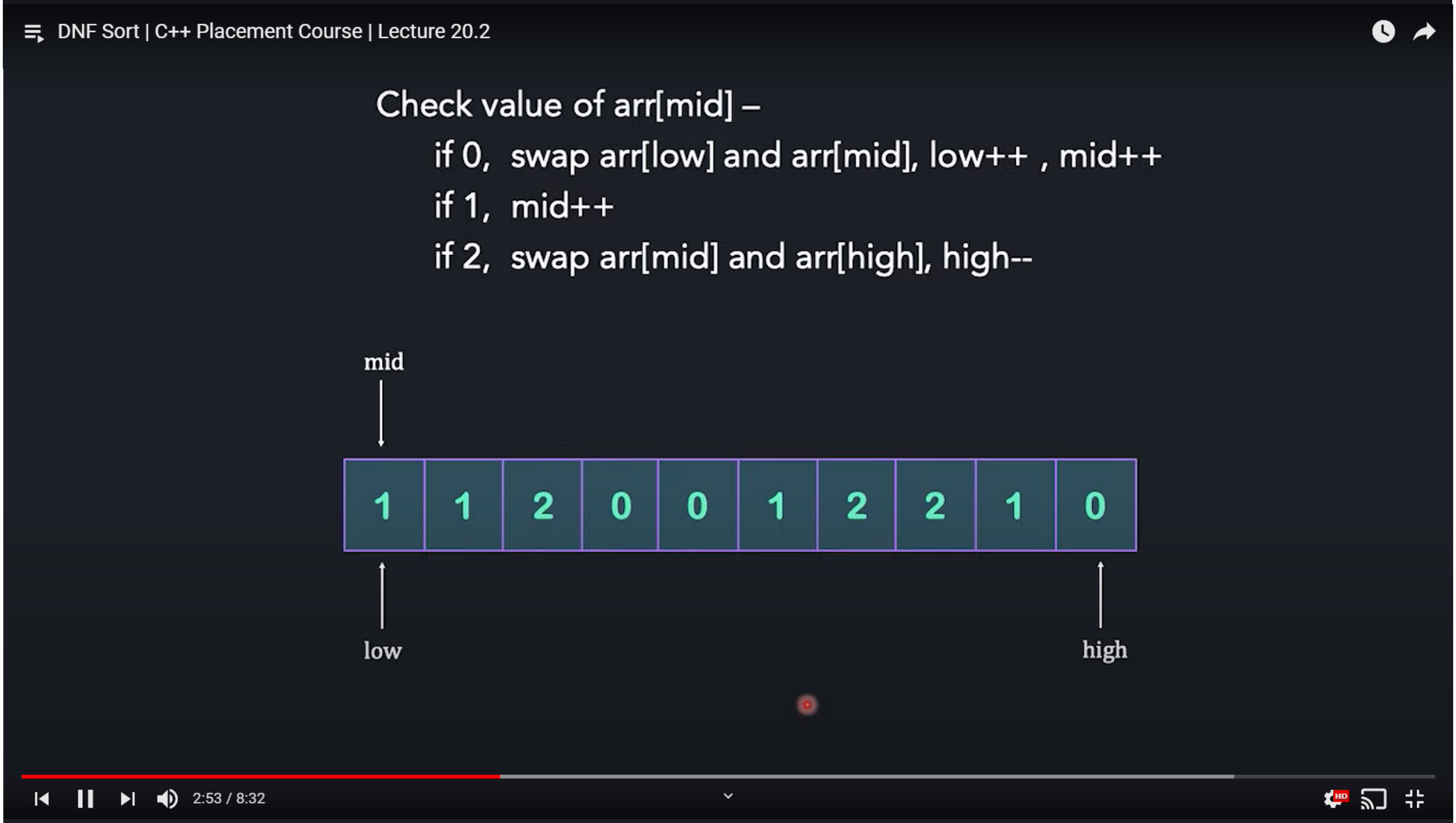
1. DNF Sort



DNF Sort | C++ Placement Course | Lecture 20.2

Check value of arr[mid] –

if 0, swap arr[low] and arr[mid], low++ , mid++
if 1, mid++
if 2, swap arr[mid] and arr[high], high--

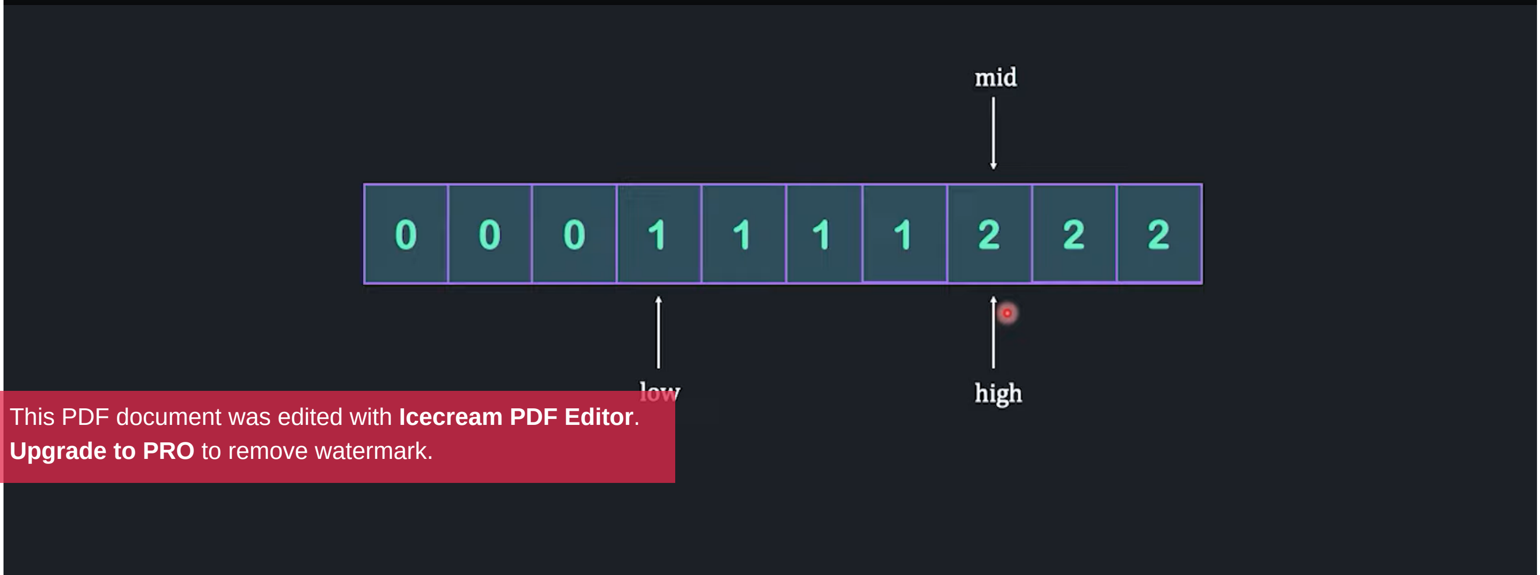


DNF Sort | C++ Placement Course | Lecture 20.2

Check value of arr[mid] –

if 0, swap arr[low] and arr[mid], low++ , mid++
if 1, mid++
if 2, swap arr[mid] and arr[high], high--

1	1	2	0	0	1	2	2	1	0



0	0	0	1	1	1	1	2	2	2