



D.N.F.(Dutch National Flag) Sort

Tags	MyNotes	Sheet	Sorting	algorithm	classic problem		
Companies	Adobe	Amazon	Goldman Sachs	Hike	MakeMyTrip	Microsoft	
	OYO	Ola Cabs	Paytm	Qualcomm	Snapdeal	Stanley	Walmart
	Yatra.com	flipkart					
Created time	@October 10, 2022 10:05 PM						
Last edited time	@October 16, 2022 7:39 AM						
URL							

Definition([source](#)):

Rearrange elements in an array into three groups: bottom, middle, and top.

One algorithm is to have the top group grow down from the top of the array, the bottom group grow up from the bottom, and keep the middle group just above the bottom. The algorithm stores the locations just below the top group, just above the bottom, and just above the middle in three indexes. At each step, examine the element just above the middle. If it belongs to the top group, swap it with the element just below the top. If it belongs in the bottom, swap it with the element just above the bottom. If it is in the middle, leave it. Update the appropriate index. Complexity is $\Theta(n)$ moves and examinations.

my def - Sorting algo for array with least to most three unique value. Space O(1), Time O(n)

in this sorting we will take three pointer(i.e. low mid and high) and use them to divide given array in four distinct part with unique values.

next we will declare and initialize two variables low and mid as 0 and array.size()-1 respectively

partition_1 - arr[0] to arr[low-1] will be for unique-Val.0(uv0)

partition_2 - arr[low] to arr[mid-1] will be for unique-Val.1(uv1)

partition_3 - arr[mid] to arr[high] will be unknown(uk)

partition_4 - arr[high+1] to arr[n] will be for unique-Val.2(uv2)

so we will get part 1 2 and 4 will be empty and part 3 will be full array of unknowns

and our task will be to eliminate all the unknowns and distribute known elements to their respective part by iterating mid 0 to size-1 with help of while($mid \leq high$) loop.

And inside that loop we will have 3 if-elif-else conditions which are

condition one - whenever we find uv1 we will just increment mid to one so partition 2 will get uv1 and unknown reign will decrease

condition two - and whenever we find uv0 we will swap it to arr[low] and incriminate low and mid to one so we will get a uv0 element in partition 1 and uv1 in partition 2 since we are iterating mid from zero and sending uv1 in partition 2 we will always get uv1 after the swap

condition three - and then last check if element is uv2 swap it to arr[high] and decrement high to one so we will get uv2 in partition 4 and since we don't know what was in arr[high] before the swap we will keep mid to its value to it will get checked in next iteration and for conditions since we have already checked for uv 1 and 0 we will guaranteed uv2 so just else will be enough but since we are using if-elseif order doesn't matter and just for code-readability we will read for uv0 and uv2 and take just else for uv1.

since we are either incrementing mid one or decrementing high one every iteration we will eventually get mid higher than high pointer and partition 3 will be empty which means we don't have any unknown valued element so we will end the loop.

for example we will take an array of ZERO's ONE's and TWO's. (0 2 1 2 0 1)

```
step.0 start
step.1 low,mid = 0 high = size-1
step.2 <arr[mid] is 0> {swap(arr[low],arr[mid])low++ mid++}
step.3 <arr[mid] is 2> {swap(arr[low],arr[hig])high--}
step.4 <arr[mid] is 1> {mid++}
step.5 <arr[mid] is 1> {mid++}
step.6 <arr[mid] is 2> {swap(arr[low],arr[hig])high--}
step.7 <arr[mid] is 0> {swap(arr[low],arr[mid])low++ mid++}
step.8 end
```

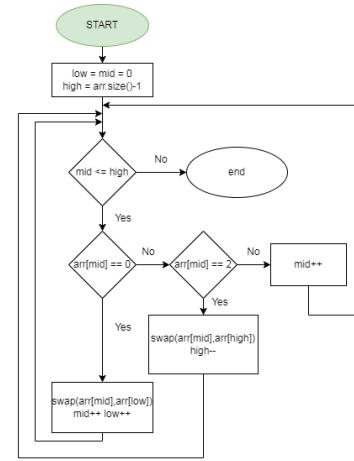
lm	h
0 2 1 2 0 1	
lm	h
0 2 1 2 0 1	
lm	h
0 1 1 2 0 2	
l m	h
0 1 1 2 0 2	
l m h	
0 1 1 0 2 2	
l h m	
0 0 1 1 2 2	

```
void sort012(int a[], int n)
{
    int low = 0;
    int mid = 0;
    int high = n-1;
    while(mid<=high)
    {
        //checking if element is 0
        if(a[mid] == 0)
```

```

        swap(a[low++],a[mid++]);
        //checking if element is 2
        else if(a[mid] == 2)
            swap(a[high--],a[mid]);
        //since we only have three unique values
        //we don't necessarily have to check for mid is 1 or not
        else
            mid++;
    }
}

```



▼ Resources i found useful

Dutch national flag

(classic problem) Definition: Rearrange elements in an array into three groups: bottom, middle, and top. One algorithm is to have the top group grow down from the top of the array, the bottom group grow up from the bottom, and keep the middle group just above the bottom.

<https://linux.nist.gov/dads/HTML/DutchNationalFlag.html>

Sort 012 | Time and Space | Data Structure and Algorithms in JAVA

G <https://www.youtube.com/watch?v=MbV26HWqxrs>



Sort an array of 0s, 1s and 2s | Dutch National Flag problem - GeeksforGeeks

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles, quizzes and practice/competitive programming/company interview Questions.

🔗 <https://www.geeksforgeeks.org/sort-an-array-of-0s-1s-and-2s/>

