

Day_16: DSA

Day-16 LeetCode: 3375 Minimum Operations to make
Array value to k

given an integer array nums and an integer k

An integer k is valid if all values in the array that are strictly
greater than k are identical

- For each index i where $\text{nums}[i] > k$, set $\text{nums}[i]$ to k

Return minimum number of operations

Code:

class Solution:

```
def minOperations (self, nums: List[int], k: int) -> int:
```

```
    numSet = set(nums)
```

```
    mn = min (nums)
```

```
    if mn < k:
```

```
        return -1
```

```
    if mn > k:
```

```
        return len (numSet)
```

```
    return len (numSet) - 1
```


SQE: • Merge two Sorted Arrays without Extra Space

given: $\text{arr1}[] \rightarrow n$
 $\text{arr2}[] \rightarrow m$ } Merge in sorted

$\text{arr1}[] = [1, 4, 8, 10]$

$\text{arr2}[] = [2, 3, 9]$

Output:

$\text{arr1}[] = [1, 2, 3, 4]$

$\text{arr2}[] = [8, 9, 10]$

⇒ Brute force:

code: `def merge(arr1, arr2, n, m):`

`# declare a 3rd array and 2 pointer`

`arr3 = [0] * (n + m)`

`left = 0`

`right = 0`

`index = 0`

`# Insert element in 3rd array`

`while left < n and right < m:`

`if arr1[left] <= arr2[right]`

`arr3[index] = arr1[left]`

`left += 1`

`index += 1`

`else:`

`arr3[index] = arr2[right]`

`right += 1`

`index += 1`

`# If left reaches end`

`while left < n:`

`arr3[index] = arr1[left]`

`left += 1`

`index += 1`

`# If right reaches end`

`while right < m:`

`arr3[index] = arr2[right]`

`right += 1`

`index += 1`

fill back the element in arr1, arr2

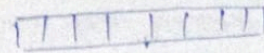
for i in range $(n+m)$:

if $i < n$:

arr1[i] = arr3[i]

else:

arr2[i-n] = arr3[i]



(n, m)

T.C $\Rightarrow O(n+m) + O(n+m)$

S.C $\Rightarrow O(n+m)$

Optimal Approach (without using any extra space)

def merge(arr1, arr2, n, m):

declare two pointer

left = n-1

right = 0

Swap the element until arr1[left] is smaller than arr2

while left ≥ 0 and right $< m$:

if arr1[left] $>$ arr2[right]

arr1[left], arr2[right] = arr2[right], arr1[left]

left -= 1

right += 1

else:

break

arr1.sort()

arr2.sort()

T.C: $O(\min(n+m) + \text{sorting})$

Sorting:

S.C = $O(1)$