

Doubt Session

1. Diversify the array.

There are k operations only.

Each decrement of a duplicate gives a chance to introduce a new unique.

Max. new unique elements you can add = $\min(k, \text{no. of duplicates})$

Total pairs in N elements, $nC_2 = \frac{n \times (n-1)}{2}$

Exa :-

Arr : 1 2 5 3

Freq : 2 3 6 7

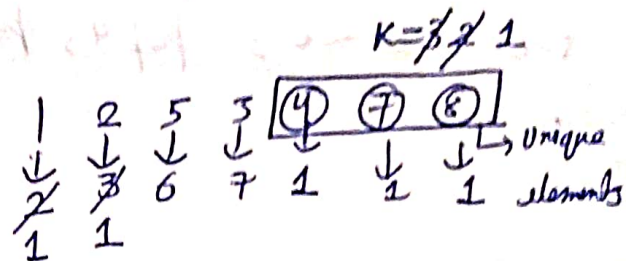
(2,2), (3,3) & (4,4) \rightarrow cannot contribute to diversity.

Out of k operations we can dec. k from existing duplicates and add to unique.

Total no. of pairs - duplicates

$nC_2 - D$

Total pairs = $18C_2 - 6C_2 - 7C_2$



1 \rightarrow 1

2 \rightarrow 1

5 \rightarrow 6 $\rightarrow 6C_2$

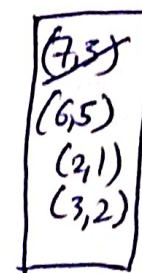
3 \rightarrow 7 $\rightarrow 7C_2$

4 \rightarrow 1

7 \rightarrow 1

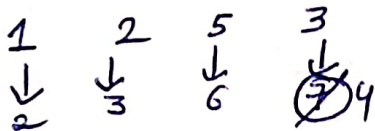
8 \rightarrow 1

Max. has P&A.



Instead of target min freq element, target max. freq element.

$k = 5$



$7-3=4$

$\Rightarrow 18C_2 - 2C_2 - 3C_2 - 4C_2 - 6C_2$

1. Use max priority queue pq (max, element).

2. Height of Soldiers

Approach :-

1. Use a stack to keep indexes of array elements. It will be a monotonic inc. stack.

2. For each element at index i :

While stack not empty & $\text{arr}[i] < \text{arr}[\text{stack}[\text{top}]]$

\Rightarrow pop from stack

Push i to the stack

3. Find min in subarray

3. Each time you pop, ^{calculate} the distance where the element is min.

4. Repeat for each window/distance.

Ex: Array : 1 5 2 9 6 3

S-1: stack [0]

S-2: $5 > 1$, push $\Rightarrow [0, 1]$

S-3: $2 < 5$, pop 1, push 2 $\Rightarrow [0, 2]$

S-4: $9 > 2$, push $\Rightarrow [0, 2, 3]$

S-5: $6 < 9$, pop 3, push 4 $\Rightarrow [0, 2, 4]$

S-6: $3 < 6$, pop 4, push 5 $\Rightarrow [0, 2, 5]$