



Subarrays with K different Integers

arr = [1 2 1 3 4] K = 3





Subarrays with K different Integers

arr = [1 2 1 3 4] K = 3

↓
1, 2





Subarrays with K different Integers

arr = [1 2 1 3 4] K = 3

Hand-drawn red arrows and brackets illustrating subarrays of the array [1, 2, 1, 3, 4] that contain exactly 3 different integers. The subarrays are [1, 2, 1], [2, 1, 3], [1, 3, 4], and [2, 1, 3, 4].





Subarrays with K different Integers

arr = [1 2 1 3 4] $K = 3$

fun (i = 0 \rightarrow n)
{

 mpp
 fun (j = i \rightarrow n)





```
for ( i = 0 → n )  
{
```

```
    mpp  
    for ( j = i → n )  
    {
```

```
        mpp[nums[j]]++;
```

```
        if ( mpp.size() == 1 ) cnt = cnt + 1
```

```
        else if ( mpp.size() > 1 ) break;
```



arr = [1 2 1 3 4] k = 3

```
    cnt = 0
    for ( i = 0 → n )
    {
        mpp
        for ( j = i → n )
        {
            mpp [ nums [ j ] ] ++ ;

            if ( mpp.size ( ) == k ) cnt = cnt + 1
            else if ( mpp.size ( ) > k ) break ;
        }
    }
    return cnt ;
```





arr = [1 2 1 3 4] k = 3

```
    cnt = 0
    for ( i = 0 → n )
    {
        mpp
        for ( j = i → n )
        {
            mpp [ nums [ j ] ] ++ ;

            if ( mpp.size ( ) == k ) cnt = cnt + 1
            else if ( mpp.size ( ) > k ) break ;
        }
    }
    return cnt ;
```

TC → $O(N^2)$

SC → $O(N)$





Subarrays with K different Integers

arr = [2 1 1 1 3 4 3 2] k = 3

N^2 \rightarrow N



Subarrays with K different Integers

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

\downarrow \downarrow

3	→	1
1	→	2
2	→	1

(num, freq)



Subarrays with K different Integers

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

$$cnt = 1$$
$$\begin{bmatrix} 4 \rightarrow 1 \\ 3 \rightarrow 1 \\ 1 \rightarrow 2 \end{bmatrix}$$

$(num, freq)$



Subarrays with K different Integers

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

cnt = 1 2 3

$$\begin{bmatrix} 4 \rightarrow 1 \\ 3 \rightarrow 2 \\ 1 \rightarrow 2 \end{bmatrix}$$

$(num, freq)$



Subarrays with K different Integers

arr = [2, 1, 1, 3, 4, 3, 2] k = 3

cnt = 1 2 3

$$\begin{bmatrix} 4 \rightarrow 1 \\ 3 \rightarrow 2 \\ 1 \rightarrow 3 \end{bmatrix}$$

$(num, freq)$





Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2 1 1 1 3 4 3 2] k = 3



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2 1 1 1 3 4 3 2] k = 3
 l r

cnt = 0

1	1
2	1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 3, 4, 3, 2] k = 3
 1 2

cnt = 0 + 3

1, 2
2, 1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

1 2

cnt = 0 + 3

1, 2
2, 1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

(Note: In the original image, the subarray [2, 1, 1, 1] is highlighted with a red box and a checkmark, with indices 1 and 3 written below it.)

$$cnt = 0 + 3 + 6$$

1, 3
2, 1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

1 2

cnt = 0 + 8 + 10

3	1
1	2
2	1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [1, 1, 1, 3] 4 3 2] $K = 3$

cnt = 0 + 3 + 10

3	1
1	3
2	1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 1, 3, 4] 3 2] $k=3$

cnt = 0 + 3 + 5 + 15

4	1
3	1
1	3
2	1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2, 1, 1, 1, 3, 4, 3, 2] k = 3

X
┌───────────┐
└───────────┘
└───────────┘
└───────────┘

cnt = 0 + 3 + 5 + 15

4, 1
3, 1
1, 3
1

(num, freq)



Subarrays with K different Integers

no. of subarrays
where different Integers $\leq K$

arr = [2 1 1 1 3 4 3 2] k=3

(Handwritten annotations: 'X' above the first '1', '1' below the first '1', '4' below the '4', and '2' below the last '2')'

cnt = 0 + 3 + 5 + 8 + 15 = 20

4	1
3	1
1	3

(num, freq)





```
func(nums, k)
```

```
{
```

```
    l = 0, r = 0, cnt = 0, mpp
```

```
    while (r < n)
```





```
mpp[nums[l]] ++;
```

```
while(mpp.size() <= K)  
{
```

```
    mpp[nums[l]] --; l = l - 1;
```





```
func (nums, k)
```

```
{  
    l = 0, r = 0, cnt = 0, mpp
```

```
while (r < nums.size)  
{
```

```
    mpp[nums[r]]++;
```

```
while (mpp.size() >= k)  
{
```

```
    mpp[nums[l]]--
```





```
map.erase(nums[l]);
```

```
l = l + 1;
```

```
}
```

```
cnt = cnt + (r - l + 1);
```

```
r = r + 1;
```

```
}
```

```
return cnt;
```





$= = k$



$$[<= k] - [<= k-1]$$

$k = 1$

$\text{func}(\text{nums}, k)$
└

$l = 0, r = 0, \text{cnt} = 0, \text{mpp}$

$\text{while} (r < \text{nums.size})$
└





$= k$



$[<= k] - [<= k-1]$

$func(nums, k) - func(nums, k-1)$

$k=1$

$func(nums, k)$
{

$l=0, r=0, cut=0, mpp$

while ($r < nums.size$)
{

$map[nums[r]]++;$



```
func(nums, k)
```

```
{  
    l = 0, r = 0, cnt = 0, mpp
```

```
while (r < nums.size) N  
{
```

```
    mpp[nums[r]]++;
```

```
while (mpp.size() <= k) → N  
{
```

```
    mpp[nums[l]]--;
```

```
    if (mpp[nums[l]] == 0)  
        mpp.erase(nums[l]);
```

```
    l = l + 1;
```

```
}
```



```

func(nums, k)
{
    l = 0, r = 0, cnt = 0, mpp
    while (r < nums.size)      N
    {
        mpp[nums[r]]++;

        while (mpp.size() <= k) → N
        {
            mpp[nums[l]]--;

            if (mpp[nums[l]] == 0)
                mpp.erase(nums[l]);

            l = l + 1;
        }
    }
}

```

$TC \rightarrow O(2N)$
 $SC \rightarrow O(N)$





$$\underline{\text{func}}(\text{nums}, K) - \underline{\text{func}}(\text{nums}, K-1)$$

$\text{func}(\text{nums}, K)$
{

$l = 0, r = 0, \text{cut} = 0, \underline{\text{mpp}}$

$2 \times$
TC $\rightarrow O(2N)$
SC $\rightarrow O(N)$

while ($r < \text{nums.size}$) N
{

$\text{mpp}[\text{nums}[r]]++;$

while ($\underline{\text{mpp.size}}() \leq K$) $\rightarrow N$
{

$\text{mpp}[\text{nums}[l]]--;$

if ($\text{mpp}[\text{nums}[l]] == 0$)
 $r = r + 1, l = l + 1$

