

Today's Content:

a. Distinct numbers in window

b. No: of distinct 2D points

c. Class Object as key → Pure Syntan Java/ Python

Q1: Given an arr[N], calculate no. of distinct elements in every subarray of size k //

Ex: 0 1 2 3 4 5 6 7
k=4 a[8] = { 2 4 3 8 3 9 9 3 }

Output: Value:

[0 3] 4

[1 4] 3

[2 5] 3

[3 6] 3

[4 7] 2

Idea1:

1. For all subarray of len k:

Insert k elements in hashset & get distinct elements

TC: $O(N-k+1) * O(k)$ SC: $O(k)$

↳ $k = N/2 = O(N^2)$ SC: $O(N)$

$k = 1$

$k = N$

Idea2:

0 1 2 3 4 5 6 7
 $k=4$ $a[s] = \{2, 4, 3, 8, 3, 9, 9, 3\}$

s	e	remove	add	HashSet	Size
0	3			{2, 4, 3, 8}	4
1	4	arr[0]	arr[4]	{ 2 , 4, 3, 8}	3
✓ 2	5	arr[1]	arr[5]	{ 4 , 3, 8, 9}	3
3	6	arr[2]	arr[6]	{ 3 , 8, 9}	2, wrong?
4	7				It cannot store frequency of elements.

Issue: Goto hashmap to store freq as well

0 1 2 3 4 5 6 7
 $k=4$ $a[s] = \{2, 4, 3, 8, 3, 9, 9, 3\}$

s	e	remove	add	HashMap <arr[i], freq>	ans
0	3			{<2, 1> <4, 1> <3, 1> <8, 1>}	4
1	4	arr[0]	arr[4]	{< 2 , 0> <4, 1> <3, 2> <8, 1>}	3
2	5	arr[1]	arr[5]	{< 4 , 0> <3, 2> <8, 1> <9, 1>}	3
3	6	arr[2]	arr[6]	{<3, 1> <8, 1> <9, 2>}	3
4	7	arr[3]	arr[7]	{<3, 2> < 8 , 0> <9, 2>}	2

void subfreq(int arr[], int k) { TC: O(N) SC: O(N)

int N = arr.length;

HashMap<Integer, Integer> hm;

/* Step 1: Insert 1st subarray [0 k-1]

for (int i = 0; i < k; i++) {

if (hm.containsKey(arr[i]) { // Inc freq by 1

int val = hm.get(arr[i])

hm.put(arr[i], val+1)

} else { hm.put(arr[i], 1) }

}

print(hm.size());

/* Step 2: Iterate all other subarrays

int s = 1, e = k;

while (e < N) {

// Remove arr[s-1]

int val = hm.get(arr[s-1])

hm.put(arr[s-1], val-1);

if (hm.get(arr[s-1]) == 0) { hm.remove(arr[s-1]) }

// add arr[e]

if (hm.containsKey(arr[e]) { // Inc freq by 1

int val = hm.get(arr[e])

hm.put(arr[e], val+1)

} else { hm.put(arr[e], 1) }

print(hm.size());

s = s+1, e = e+1;

}

}

Q2: Given N points on a 2d plane, in a 2D matrix
Return total no: of distinct points

Ex: N=8 mat[8][2]

	0:x	1:y		
0	5	6	✓	<u>Distinct Points = 5</u>
1	2	8	✓	Idea1:
2	-1	-1	✓	a) Insert in hashmap <2,6> <2,8> ✗
3	2	-3	✓	<2,8>
4	2	8	✗	b) Concatenate x & y as string and Insert ✗
5	7	7	✓	Ex: <21,3> → 213
6	2	8	✗	<2,13> → 213
7	2	-3	✗	c) Concatenate x & y as string with separator and Insert
				Ex: <21,3> Insert # : 21#3 } different ✓
				<2,13> Insert # : 2#13 } strings ✓

```

int distinctPoints(int mat[][2]) {
    int N = mat.length;
    HashSet<String> hs = new HashSet<>();
    for (int i = 0; i < N; i++) {
        // Insert each point in hashset
        int x = mat[i][0], y = mat[i][1];
        String s = x + "#" + y;
        hs.add(s);
    }
    return hs.size();
}

```

Object as key:

1