Count Distinct element from diff. window

arr[] = {10, 20, 10, 10, 30, 40}

first window

① {10, 20, 10, 10, 30, 40}

Create a hash map containing it value and associated frequency

2 distinct value

Value	frequency
10	2
20	1

removed the first element from previous window

2nd window

② {10, 20, 10, 10, 30, 40}

2 distinct value

update the hash map

Value	frequency
20	1
10	2

removed the first element from previous window

3rd window

③ {10, 20, 10, 10, 30, 40}

Update the hash map

2 distinct element

Value	frequency
10	2
30	1

4

{10, 20, 10, 10, 30, 40}

4th window

remove the first element from previous window

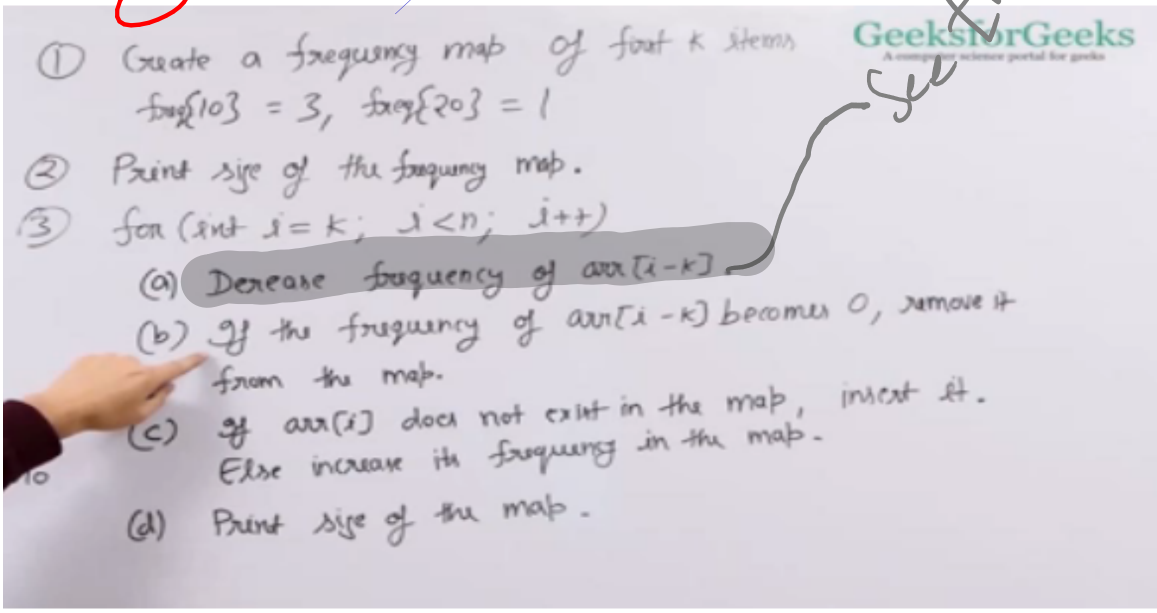
Update the hash map

3 distinct element

Value	Frequency
10	~~2~~ 2
30	1
40	1

See this step

Algorithm



① Create a frequency map of first k items
freq{10} = 3, freq{20} = 1

② Print size of the frequency map.

```
③ for (int i = k; i<n; i++)
    (a) Decrease frequency of arr[i-k]
    (b) If the frequency of arr[i-k] becomes 0, remove it
        from the map.
    (c) If arr[i] does not exist in the map, insert it.
        Else increase its frequency in the map.
    (d) Print size of the map.
```

Code

```
#include <bits/stdc++.h>
using namespace std;

void printDistinct(int arr[], int n, int k)
{
    map<int, int> m;

    for (int i = 0; i < k; i++) {
        m[arr[i]] += 1;
    }

    cout << m.size() << " ";

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

        m[arr[i - k]] -= 1;

        if (m[arr[i - k]] == 0) {
            m.erase(arr[i-k]);
        }
        m[arr[i]] += 1;

        cout << m.size() << " ";
    }

}

int main()
{
    int arr[] = {10, 10, 5, 3, 20, 5};

    int n = sizeof(arr)/sizeof(arr[0]);
    int k=4;
    printDistinct(arr, n, k);
}
```