

## Distinct Elements in each Window in C++

```
#include <iostream>
#include <unordered_map> // for unordered_map

using namespace std;

void printDistinct(int arr[], int n, int k) {
    unordered_map<int, int> m; // Declaration of
    unordered_map to store element frequencies

    // Count frequencies of first window
    for (int i = 0; i < k; i++) {
        m[arr[i]]++;
    }

    // Print the size of the map for the first window
    cout << m.size() << " ";

    // Process subsequent windows
    for (int i = k; i < n; i++) {
        // Remove the element that is moving out of the
        window
        m[arr[i - k]]--;

        // Remove the element from map if its count
        becomes zero
        if (m[arr[i - k]] == 0) {
            m.erase(arr[i - k]);
        }

        // Add the new element to the map
        m[arr[i]]++;

        // Print the size of the map for the current
        window
        cout << m.size() << " ";
    }
}

int main() {
    int arr[] = {10, 10, 5, 3, 20, 5};
    int n = sizeof(arr) / sizeof(arr[0]); // Calculate the
    size of the array
    int k = 4; // Size of the window

    // Call the function to print distinct elements in
    every window of size k
    printDistinct(arr, n, k);

    cout << endl;

    return 0;
}
```

### Input:

Array: {10, 10, 5, 3, 20, 5}  
Window size k = 4

### Step 1: Initialize the First Window (size k)

The function first processes the first k elements of the array and calculates their frequencies using an unordered map.

#### First Window (arr[0..3] = {10, 10, 5, 3}):

- m[10]++ → {10: 2}
- m[5]++ → {10: 2, 5: 1}
- m[3]++ → {10: 2, 5: 1, 3: 1}

Distinct elements count = **3** (keys: 10, 5, 3).

### Output so far:

3

### Step 2: Slide the Window

Now, we slide the window over the array. For each new element that comes into the window, we remove the element that goes out of the window and update the frequencies accordingly.

#### Second Window (arr[1..4] = {10, 5, 3, 20}):

1. Remove the element arr[0] (10) that goes out of the window:
  - m[10]-- → {10: 1, 5: 1, 3: 1}
  - Since m[10] == 0, remove 10 from the map.
  - Map becomes: {5: 1, 3: 1}
2. Add the new element arr[4] (20):
  - m[20]++ → {5: 1, 3: 1, 20: 1}

Distinct elements count = **4** (keys: 5, 3, 20).

### Output so far:

3 4

#### Third Window (arr[2..5] = {5, 3, 20, 5}):

1. Remove the element arr[1] (10) that goes out of the window:
  - m[10]-- → {5: 1, 3: 1, 20: 1} (no change since 10 was already removed).
2. Add the new element arr[5] (5):

	<div>○ m[5]++ → {5: 2, 3: 1, 20: 1}</div> <div>Distinct elements count = <b>3</b> (keys: 5, 3, 20).</div> <div><b>Output so far:</b></div> <div>3 4 3</div>
<div>Output:</div> <div>3 4 3</div>	