|  |  |
| --- | --- |
| **Distinct Elements Window of Size K in C++** | |
| #include <iostream>  #include <vector>  #include <unordered\_map>  #include <deque>  using namespace std;  vector<int> distinctElementsInWindow(const vector<int>& arr, int k) {  vector<int> result;  unordered\_map<int, int> frequencyMap;  int n = arr.size();  int i = 0;  // Initialize the frequency map for the first window  for (i = 0; i < k - 1; ++i) {  frequencyMap[arr[i]]++;  }  for (int j = -1; i < n; ++i, ++j) {  // Add the next element (i-th element) to the frequency map  frequencyMap[arr[i]]++;  // Record the number of distinct elements in the current window  result.push\_back(frequencyMap.size());  // Remove the (j-th element) as the window slides  if (j >= 0) {  if (frequencyMap[arr[j]] == 1) {  frequencyMap.erase(arr[j]);  } else {  frequencyMap[arr[j]]--;  }  }  }  return result;  }  int main() {  vector<int> arr = {2, 5, 5, 6, 3, 2, 3, 2, 4, 5, 2, 2, 2, 2, 3, 6};  int k = 4;  vector<int> result = distinctElementsInWindow(arr, k);  for (int num : result) {  cout << num << " ";  }  cout << endl;  return 0;  } | Dry Run:Initialize:  * **arr = [2, 5, 5, 6, 3, 2, 3, 2, 4, 5, 2, 2, 2, 2, 3, 6]** * **k = 4** * **frequencyMap** = {} (Empty at the start) * **result** = [] (Empty at the start)  Step-by-Step Iteration:  | **i** | **arr[i]** | **frequencyMap (Updated)** | **Distinct Elements** | **result (after update)** | **j** | | --- | --- | --- | --- | --- | --- | | 0 | 2 | {2: 1} | 1 | [] | -1 | | 1 | 5 | {2: 1, 5: 1} | 2 | [] | 0 | | 2 | 5 | {2: 1, 5: 2} | 2 | [] | 1 | | 3 | 6 | {2: 1, 5: 2, 6: 1} | 3 | [3] | 2 | | 4 | 3 | {2: 1, 5: 1, 6: 1, 3: 1} | 4 | [3, 4] | 3 | | 5 | 2 | {2: 2, 5: 1, 6: 1, 3: 1} | 4 | [3, 4, 4] | 4 | | 6 | 3 | {2: 2, 5: 1, 6: 1, 3: 2} | 3 | [3, 4, 4, 3] | 5 | | 7 | 2 | {2: 3, 5: 1, 6: 1, 3: 2} | 3 | [3, 4, 4, 3, 3] | 6 | | 8 | 4 | {2: 3, 5: 1, 6: 1, 3: 2, 4: 1} | 4 | [3, 4, 4, 3, 3, 4] | 7 | | 9 | 5 | {2: 3, 5: 2, 6: 1, 3: 2, 4: 1} | 4 | [3, 4, 4, 3, 3, 4, 4] | 8 | | 10 | 2 | {2: 4, 5: 2, 6: 1, 3: 2, 4: 1} | 3 | [3, 4, 4, 3, 3, 4, 4, 3] | 9 | | 11 | 2 | {2: 5, 5: 2, 6: 1, 3: 2, 4: 1} | 2 | [3, 4, 4, 3, 3, 4, 4, 3, 3] | 10 | | 12 | 2 | {2: 6, 5: 2, 6: 1, 3: 2, 4: 1} | 1 | [3, 4, 4, 3, 3, 4, 4, 3, 3, 2] | 11 | | 13 | 2 | {2: 7, 5: 2, 6: 1, 3: 2, 4: 1} | 1 | [3, 4, 4, 3, 3, 4, 4, 3, 3, 2, 2] | 12 | | 14 | 3 | {2: 7, 5: 2, 6: 1, 3: 3, 4: 1} | 2 | [3, 4, 4, 3, 3, 4, 4, 3, 3, 2, 2, 3] | 13 | | 15 | 6 | {2: 7, 5: 2, 6: 2, 3: 3, 4: 1} | 3 | [3, 4, 4, 3, 3, 4, 4, 3, 3, 2, 2, 3, 3] | 14 |  Final Result: The output is the list of distinct elements in each sliding window of size k as the window slides across the array:  **Output:**  3 4 4 4 3 3 4 4 3 3 2 2 3 |
| Output: 3 4 4 4 3 3 4 4 3 3 2 2 3 | |