```
// Given two arrays: a1[0..n-1] of size n and a2[0..m-1] of
size m. Task is to check whether a2[] is a subset of a1[] or
not. Both the arrays can be sorted or unsorted.
// Expected Time Complexity: O(n)
// Expected Auxiliary Space: O(n)

// Test cases -

// o/p: no👇
    int a1[] = {1, 2, 3, 4, 5, 6, 7, 8};
    int a2[] = {1, 2, 3, 1};

// o/p: yes👇
    int a1[] = {1, 2, 3, 4, 5, 1, 1, 1};
    int a2[] = {1, 2, 3, 1};
```

a2 array is a subset of a1 array if all the elements of a2 are in a1 also. For a repeated element, it's occurrences in a1 should be greater than or equal to that in a2 array.

Optimized: Define two unordered_map – 'map1' and 'map2'. 'map1' stores <no., it's no. of occurrences> of all the elements of a1 array. Similar for 'map2' as well.

Traverse 'map2'. If a key is already present in 'map1'. Compare it's occurrence in a1 with that in a2. It's occurrences in a1 should be greater than or equal to that in a2 array.