Maximum Frequency Number

Problem Description: You are given with an array of integers that contain numbers in random order. Write a program to find and return the number which occurs maximum times in the given input.
If more than one element occurs same number of times in the input, return the element which is present in the input first.

Sample Input:

```
8
32 14 14 76 14 32 29 32
```

Sample Output:

```
32
```

How to approach?

The naive approach for this problem would be to first sort the array. This would group similar numbers together and we could then count the numbers in all the groups, find the largest group and return that group's number. If there are multiple largest groups, we return the group whose number occurred first in the original array.

The time complexity of this solution would be O(n * log(n)) since we are sorting the array and that is the most expensive operation (time-wise) that we do.

Let's think of a better approach. We want the frequency of every element without sorting the array. Also, we would like to retrieve the frequency of the element in O(1) time and even update it in O(1) time to further optimize our code. How can we achieve that? The first data structure that comes to mind is a HashMap.
We could iterate through the array, increase the frequency of each element by 1 as we encounter it (initially the frequency would be 0). Also, we could maintain the maximum frequency as we iterate through the array.

Now we would know the maximum frequency but we still don’t know which element that frequency belongs to. There could be multiple elements with the same frequency.
So, we iterate through the array once more and check whether the frequency of the current element is equal to the maximum frequency we found earlier. If they are equal we return that element and exit the function.

The pseudo-code for this approach is shown on the next page.

```
function maximumFrequency(arr):
  HashMap(integer,integer) frequencyMap
  maxFrequency = -1
  for element in arr:
    //frequencyMap[element] is initially 0
    frequencyMap[element] <- frequencyMap[element] + 1
    maxFrequency <- max(maxFrequency, frequencyMap[element])

  for element in arr:
    if(frequencyMap[element] = maxFrequency):
      return element
```