Pseudocode Problem 1

M = unique elements

```
def topMostKFrequent(nums, k):
    Initialize map                                         —--> O(1)
    Store the frequency of each element                        —--> O(logk)
    For number in nums:                                        —--> O(n)
        Increment the frequency if the number is in the map  —> O(1)
    Initialize heap                                        —--> O(1)
    Add elements to the heap with for loop (for each key, value in map):      —--> O(m)
        Push the (value, key) into the heap in any order (did frequency first)—--> O(logk)
        If the heap size is greater than given k: —--> O(1)
            Remove smallest element from heap —->O(logk)
    Get the keys from the heap —--> O(k)
    Return the list of the keys —--> O(1)
```

Overall Time Complexity: O(n + m*logk) but in some cases the worst time complexity could be O(n+n*logk) since that would mean all of the elements in the given list are unique.