

Top K frequent elements

```
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        int i;
        vector<int> awn;
        unordered_map<int, int> map;
        int cur_top;
        stack<int> st;

        for (i = 0; i < nums.size(); i++) {
            map[nums[i]] += 1;
        }

        for (i = 0; i < k; i++) {
            cur_top = 0;
            st.push(0);
            for (auto i: map) {
                if (i.second > cur_top) {
                    st.pop();
                    st.push(i.first);
                    cur_top = i.second;
                }
            }
            map.erase(st.top());
            awn.push_back(st.top());
        }

        return awn;
    }
};
```

- working solution

- first goes through the array of numbers and stores frequencies of numbers in a hash map
- hash map is iterated to find the number with the highest number
 - current highest number is pushed on to a stack and popped off if surpassed in frequency
- step 2 is repeated k times