

最大频率栈

分析：

1. 每次pop需要知道最高频率元素是多少；
2. 如果最高频率元素有多个，需要知道最近push进来的是哪个元素。

为解决上述问题：

1. 需要记录当前的最高频率；
2. 对每次压入栈的数据，需要构造一个map, key为freq, val 为压入栈元素列表，构成一个stack 。

```
class FreqStack {
public:
    FreqStack() {}

    void push(int val) {
        int freq = 0;
        if (valToFreq.count(val)) {
            freq += valToFreq[val];
        }

        valToFreq[val] = freq + 1;
        freqToVals[freq + 1].push(val);
        maxFreq = maxFreq > (freq + 1) ? maxFreq : freq + 1;
    }

    int pop() {
        int v = freqToVals[maxFreq].top();
        freqToVals[maxFreq].pop();
        valToFreq[v]--;

        if (freqToVals[maxFreq].empty()) {
            maxFreq--;
        }

        return v;
    }

private:
    int maxFreq = 0; // 最大频率
    std::unordered_map<int, int> valToFreq;
    std::unordered_map<int, std::stack<int>> freqToVals;
};
```