445 & 349

Category	
Difficulty	
Note	
AC & Time	
Property	
related	

45. Add Two Number II

```
ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
        //method 1: reverse the Link-Lists O(n)  //add two numbers O(n)
        //reverse back O(n)
        // method 2: using stack // calculate //build link list O(n)
        stack<int> sl1, sl2, sadd;
        while (l1) {
            sl1.push(l1->val);
            l1 = l1->next;
        }
        while (l2) {
            sl2.push(l2->val);
            l2 = l2->next;
        }
        // add two
        int sum = 0;
        int carry = 0;
        while (!sl1.empty() || !sl2.empty()) {
            if (sl1.empty()) {
                sum = sl2.top() + carry;
                sl2.pop();
            } else if (sl2.empty()) {
                sum = sl1.top() + carry;
                sl1.pop();
            } else {
                sum = sl1.top() + sl2.top() + carry;
                sl1.pop();
                sl2.pop();
            }

            carry = sum / 10;
            sum = sum % 10;
            sadd.push(sum);
```

```
        }

        if (carry > 0) sadd.push(carry);
        ListNode* dummy = new ListNode(0);
        ListNode* cur = dummy;
        while (!sadd.empty()) {
            ListNode* add = new ListNode(sadd.top());
            sadd.pop();
            cur->next = add;
            cur = add;
        }

       return dummy->next;

    }
};
```

349. Intersection of Two Arrays

```
class Solution {
public:
    vector<int> intersection(vector<int>& nums1, vector<int>& nums2) {
        vector<int> res;
        if (nums1.size() == 0 || nums2.size() == 0) return res;
        unordered_map<int, int> dict;
        unordered_set<int> resMap; //unique
        for (auto& n1 : nums1) {
            dict[n1]++;
        }

        for (auto& n2 : nums2) {
            if (dict.count(n2) && !resMap.count(n2)){
                resMap.insert(n2);
                res.push_back(n2);
            }
        }

        return res;
    }
};

//using hash map and hash set so space is O(n)
//Time is O(n) bulid the hash map;
```