Leetcode

`IT Studies`

Easy

1. Two Sum

– map

```
public class Solution {
    public int[] twoSum(int[] nums, int target) {

        int[] result = new int[2];
        Map<Integer, Integer> map = new HashMap<Integer, Integer>
();

        for(int i = 0; i < nums.length; i++){
            if(map.containsKey(target - nums[i])){
                result[1] = i;
                result[0] = map.get(target - nums[i]);
            }
            map.put(nums[i],i);
        }
        return result;
    }
}
```

7. Reverse Integer

– %findlast, /10

```
public class Solution {
    public static int reverse(int x) {
        long result = 0;
        while(x != 0){
            result = x % 10 + result * 10;
            x /= 10;
        }
        if(result > Integer.MAX_VALUE || result < Integer.MIN_VALUE
){
            return 0;
        }
        return (int) result;
    }
}
```

9. Palindrome Number

– string -> toCharArray

```java
public class Solution {
    public boolean isPalindrome(int x) {
        if(x < 0){
            return false;
        }
        String word = Integer.toString(x);
        char[] nums = word.toCharArray();
        int l = word.length();
        for(int i = 0; i <= l/2; i++){
            if(nums[i] != nums[l-1-i]){
                return false;
            }
        }
        return true;
    }
}
```

13. Roman to Integer

– 左：小减大加

```java
public class Solution {
    public int romanToInt(String s) {
        int l = s.length();
        int result = 0;
        for(int i = 0; i < l; i++){
            int num1 = basicRomanInt(s.substring(i,i+1));
            if(i < l-1){
                int num2 = basicRomanInt(s.substring(i+1,i+2));
                if(num1 < num2){
                    num1 = -num1;
                }
            }
            result += num1;
        }
        return result;
    }

    public int basicRomanInt(String s){
        if(s.equals("I")){
            return 1;
        }
        if(s.equals("V")){
            return 5;
        }
        if(s.equals("X")){
            return 10;
        }
        if(s.equals("L")){
            return 50;
        }
        if(s.equals("C")){
            return 100;
        }
        if(s.equals("D")){
            return 500;
        }
        return 1000;
    }
}
```

14. Longest Common Prefix

–

```
class Solution {
public:
    string longestCommonPrefix(vector<string>& strs) {
        string prefix = "";
        for(int idx=0; strs.size()>0; prefix+=strs[0][idx], idx++)
            for(int i=0; i<strs.size(); i++)
                if(idx >= strs[i].size() ||(i > 0 && strs[i][idx]
!= strs[i-1][idx]))
                    return prefix;
        return prefix;
    }
};
```

20. Valid Parentheses

– replace 括号

```
public class Solution {
    public boolean isValid(String s) {
        int l = 0;

        while(l != s.length()){
            l = s.length();
            s = s.replace("()", "").replace("{}", "").replace("[]",
"");
        }

        return l == 0;
    }
}
```

21. Merge Two Sorted Lists

– Recursion: mergeTwoLists == 最小值

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        if (l1 == NULL){
            return l2;
        }
        if (l2 == NULL){
            return l1;
        }

        if (l1->val < l2->val){
            l1->next = mergeTwoLists(l1->next, l2);
            return l1;
        }
        else {
            l2->next = mergeTwoLists(l1, l2->next);
            return l2;
        }
    }
};
```

26. Remove Duplicates from Sorted Array

– 加入不同element: 后面 > 前边，则加后面的

```
public class Solution {
    public int removeDuplicates(int[] nums) {
        int l = nums.length;
        if(l < 2){
            return l;
        }

        l = 1;
        for(int i = 1; i < nums.length; ++i) {
            if(nums[i] != nums[i-1]) {
                nums[l++] = nums[i];
            }
        }
        return l;
    }
}
```

27. Remove Element

– 加入特定的element: 不等于val，则加入

```java
public class Solution {
    public int removeElement(int[] nums, int val) {
        int l = 0;
        for(int i = 0; i< nums.length; i++){
            if(nums[i] != val){
                nums[l++] = nums[i];
            }
        }
        return l;
    }
}
```