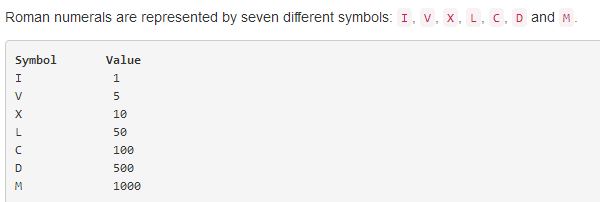
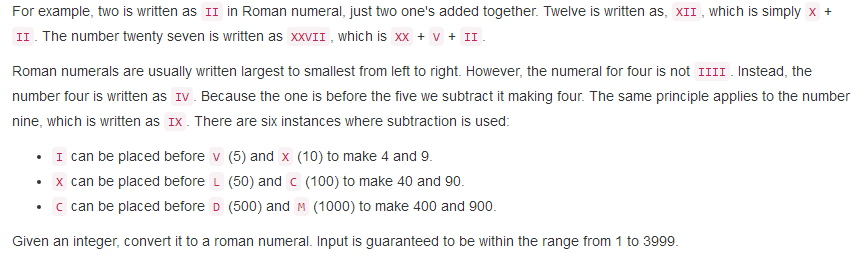
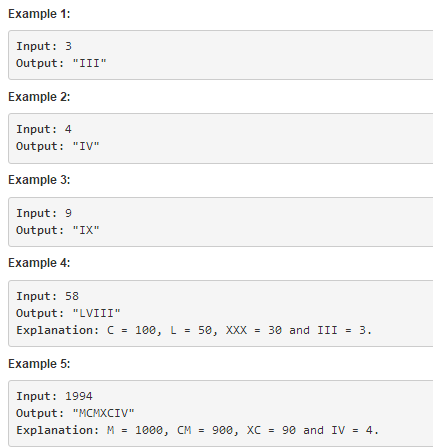
Integer to Roman and Roman to Integer

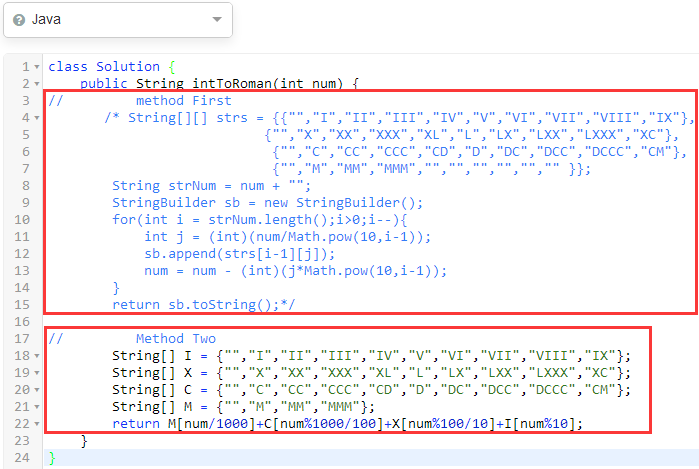
# Leetcode. 12. Integer to Roman







MyAnswer：



第二种方法战胜了95%的算法。

技巧：对于一个int数，如何获取各个位上的数字。一开始考虑先减去高位，然后再求商。

简单方法：先求余，再求商。如1368取出十位数字：1368%100/10

理解：求余作用去除高位；求商作用保留高位。

class Solution {

public String intToRoman(int num) {

// method First

/\* String[][] strs = {{"","I","II","III","IV","V","VI","VII","VIII","IX"},

{"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"},

{"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"},

{"","M","MM","MMM","","","","","","" }};

String strNum = num + "";

StringBuilder sb = new StringBuilder();

for(int i = strNum.length();i>0;i--){

int j = (int)(num/Math.pow(10,i-1));

sb.append(strs[i-1][j]);

num = num - (int)(j\*Math.pow(10,i-1));

}

return sb.toString();\*/

// Method Two

String[] I = {"","I","II","III","IV","V","VI","VII","VIII","IX"};

String[] X = {"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"};

String[] C = {"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"};

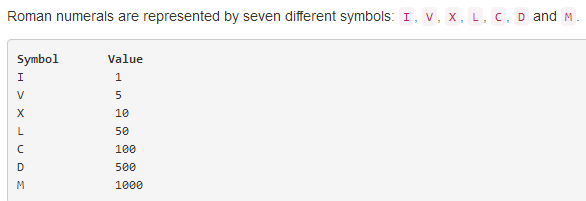
String[] M = {"","M","MM","MMM"};

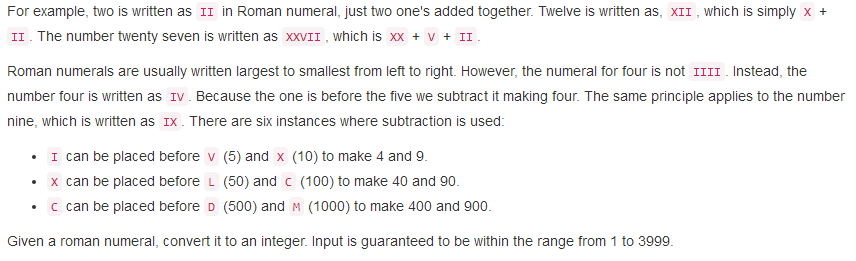
return M[num/1000]+C[num%1000/100]+X[num%100/10]+I[num%10];

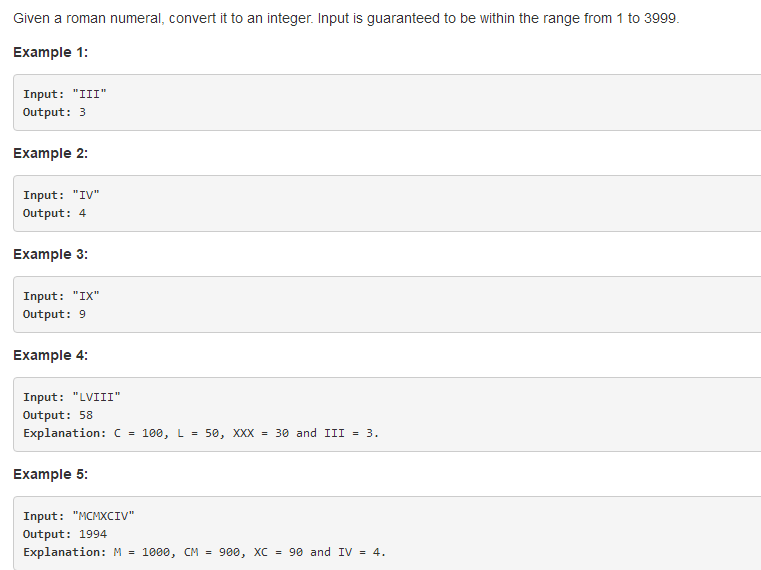
}

}

# Leetcode. 13. Roman to Integer







My Answers:

class Solution {

public int romanToInt(String s) {

// Method First

/\* if(s == null || s == "")

return 0;

String[] nums = {"1","2","3","5","4","6","7","8","9"};

String[] I = {"I","II","III","V","IV","VI","VII","VIII","IX"};

String[] X = {"X","XX","XXX","L","XL","LX","LXX","LXXX","XC"};

String[] C = {"C","CC","CCC","D","CD","DC","DCC","DCCC","CM"};

String[] M = {"M","MM","MMM"};

StringBuilder sb = new StringBuilder();

for(int i = M.length-1;i>=0;i--){

if(s.contains(M[i])&&s.startsWith(M[i])){

sb.append(nums[i]);

s = s.substring(M[i].length());

break;

}

if(i==0){

sb.append("0");

}

}

for(int i = C.length-1;i>=0;i--){

if(s==null||s=="") break;

if(s.contains(C[i])&&s.startsWith(C[i])){

sb.append(nums[i]);

s = s.substring(C[i].length());

break;

}

if(i==0){

sb.append("0");

}

}

for(int i = X.length-1;i>=0;i--){

if(s==null||s=="") break;

if(s.contains(X[i])&&s.startsWith(X[i])){

sb.append(nums[i]);

s = s.substring(X[i].length());

break;

}

if(i==0){

sb.append("0");

}

}

for(int i = I.length-1;i>=0;i--){

if(s==null||s=="") break;

if(s.contains(I[i])&&s.startsWith(I[i])){

sb.append(nums[i]);

s = s.substring(I[i].length());

break;

}

if(i==0){

sb.append("0");

}

}

int result = Integer.parseInt((sb.toString().equals("")) ? "0" : sb.toString());

return result;\*/

// Method Second

if(s == null || s == "")

return 0;

String[] nums = {"1","2","3","5","4","6","7","8","9"};

String[] I = {"I","II","III","V","IV","VI","VII","VIII","IX"};

String[] X = {"X","XX","XXX","L","XL","LX","LXX","LXXX","XC"};

String[] C = {"C","CC","CCC","D","CD","DC","DCC","DCCC","CM"};

String[] M = {"M","MM","MMM"};

StringBuilder sb = new StringBuilder();

int index = perWei(M,s);

sb.append(index == -1 ? 0 : nums[index]);

s = s.substring(index == -1 ? 0 : M[index].length());

index = perWei(C,s);

sb.append(index == -1 ? 0 : nums[index]);

s = s.substring(index == -1 ? 0 : C[index].length());

index = perWei(X,s);

sb.append(index == -1 ? 0 : nums[index]);

s = s.substring(index == -1 ? 0 : X[index].length());

index = perWei(I,s);

sb.append(index == -1 ? 0 : nums[index]);

s = s.substring(index == -1 ? 0 : I[index].length());

int result = Integer.parseInt(sb.toString());

return result;

}

private int perWei(String[] P,String s){

if(s == null||s == "")

return -1;

for(int i = P.length-1;i>=0;i--){

if(s.contains(P[i])&&s.startsWith(P[i])){

return i;

}

}

return -1;

}

}

别人的算法:js is short

/\*\*

@param {string} s

@return {number}

\*/

var romanToInt = function(s) {

var ans = 0;

var arr = ['M', 'CM', 'D', 'CD', 'C', 'XC', 'L', 'XL', 'X', 'IX', 'V', 'IV', 'I'];

var val = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1];

var len = arr.length;

var index = 0;

var num;

for(var i = 0; i < len; i ++){

num = 0;

while(s.indexOf(arr[i], index) != -1 && s.indexOf(arr[i], index) == index){

num ++;

index += arr[i].length;

}

ans += val[i] \* num;

}

return ans;

};