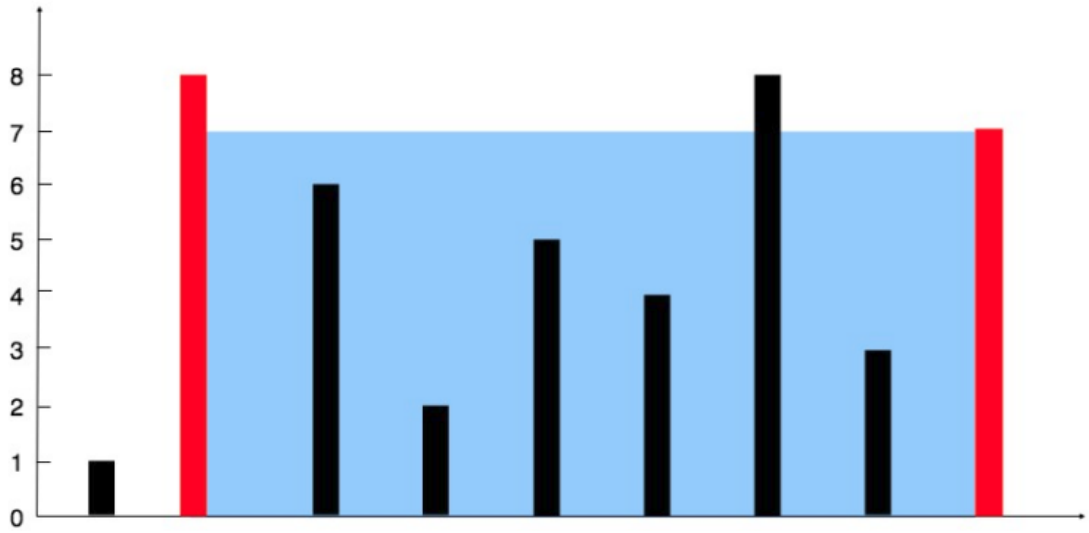
题目描述

题目：11.装最多的水

Given n non-negative integers a_1, a_2, ..., a_n, where each represents a point at coordinate (i, a_i). n vertical lines are drawn such that the two endpoints of line i is at (i, a_i) and $(i, 0)$. Find two lines, which together with x-axis forms a container, such that the container contains the most water.

Note: You may not slant the container and n is at least 2.



The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

给你一个非负的整形数组，通过柱形图的形式表示每一个数字，求出柱形图中，两条柱子能组成的最大矩形面积。

解题思路

一开始就能想到的解决方案是嵌套遍历。

但是认真观察上图，会发现面积取决于宽度和高度，高度取决于低的柱子。

代码实现

```
public int maxArea(int[] height) {
    int l = 0, r = height.length - 1;
    int area = 0;
    while (l < r) {
        area = Math.max(area, (r - l) * Math.min(height[l], height[r]));
        if (height[l] < height[r]) l++;
        else r--;
    }
    return area;
}
```

题目描述

题目：12.阿拉伯数字转罗马数字

Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, two is written as `II` in Roman numeral, just two one's added together. Twelve is written as, `XII`, which is simply `X` + `II`. The number twenty seven is written as `XXVII`, which is `XX` + `V` + `II`.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

Given an integer, convert it to a roman numeral. Input is guaranteed to be within the range from 1 to 3999.

Example 1:

```
Input: 3
Output: "III"
```

给你一个整数，转换成罗马数字的形式，整数的范围是0~3999.

解题思路

代码实现

```
public static String intToRoman(int num) {
    String M[] = {"", "M", "MM", "MMM"};
    String C[] = {"", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"};
    String X[] = {"", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"};
    String I[] = {"", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"};
    return M[num/1000] + C[(num%1000)/100] + X[(num%100)/10] + I[num%10];
}
```

题目描述

题目：13罗马数字转阿拉伯数字

Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, two is written as `II` in Roman numeral, just two one's added together. Twelve is written as, `XII`, which is simply `X` + `II`. The number twenty seven is written as `XXVII`, which is `XX` + `V` + `II`.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer. Input is guaranteed to be within the range from 1 to 3999.

Example 1:

```
Input: "III"
Output: 3
```

解题思路

代码实现

```
public int romanToInt(String s) {
  HashMap<Character, Integer> map = new HashMap<>(8);
  // @formatter:off
  map.put('I', 1);map.put('V', 5);
  map.put('X', 10);map.put('L', 50);
  map.put('C', 100);map.put('D', 500);
  map.put('M', 1000);
  // @formatter:on
  int result = map.get(s.charAt(0));
  for (int i = 1; i < s.length(); i++) {
    int prev = map.get(s.charAt(i - 1));
    int cur = map.get(s.charAt(i));
    if (prev<cur)
      result += cur - 2 * prev;
    else
      result += cur;
  }
```

```
    return result;
  }
```