compare-version-numbers

```
package algorithm.string;

/**
 * https://leetcode.com/problems/compare-version-numbers/
 *
 Compare two version numbers version1 and version2.
 If version1 > version2 return 1, if version1 < version2 return -1, otherwi

 You may assume that the version strings are non-empty and contain only dig
 The . character does not represent a decimal point and is used to separate
 For instance, 2.5 is not "two and a half" or "half way to version three",

 Here is an example of version numbers ordering:
 0.1 < 1.1 < 1.2 < 13.37

 * @author xiaobaoqiu  Date: 16-7-11 Time:  下午11:42
 */
public class CompareVersionNumbers {
    public static void main(String[] args) {
        System.out.println(compareVersion("0.1", "1.1"));//-1
        System.out.println(compareVersion("1.1", "1")); //1
        System.out.println(compareVersion("1.0", "1")); //0
    }

    /**
     * 思路：用.切割成数组，再逐个比较
     *
     * 4 ms
     * Your runtime beats 9.96% of java submissions
     */
    public static int compareVersion(String version1, String version2) {
        String[] v1s = version1.split("\\.");
        String[] v2s = version2.split("\\.");

        int i = 0;
        for (; i < v1s.length && i < v2s.length; i++) {
            int v1 = Integer.parseInt(v1s[i]);
            int v2 = Integer.parseInt(v2s[i]);
            if (v1 > v2) return 1;
            if (v1 < v2) return -1;
        }
        while (i < v1s.length) {
            int v1 = Integer.parseInt(v1s[i++]);
            if (v1 > 0) return 1;
        }
        while (i < v2s.length) {
            int v2 = Integer.parseInt(v2s[i++]);
            if (v2 > 0) return -1;
        }
        return 0;
    }
}
```

count-and-say

```
package algorithm.string;

/**
 * https://leetcode.com/problems/count-and-say/
 *
 The count-and-say sequence is the sequence of integers beginning as follow
 1, 11, 21, 1211, 111221, ...

 1 is read off as "one 1" or 11.
 11 is read off as "two 1s" or 21.
 21 is read off as "one 2, then one 1" or 1211.
 Given an integer n, generate the nth sequence.

 Note: The sequence of integers will be represented as a string.

 * @author xiaobaoqiu  Date: 16-7-12 Time:  下午9:50
 */
public class CountAndSay {
    public static void main(String[] args) {
        int n = 10;
        countAndSay(10);
    }

    /**
     * 思路：模拟
     *
     * 5 ms
     * Your runtime beats 48.34% of java submissions
     */
    public static String countAndSay(int n) {
        String s = "1";
        StringBuilder builder = new StringBuilder();
        for (int i = 2; i <= n; i++) {
            char pre = s.charAt(0);
            int count = 1;
            for (int j = 1; j < s.length(); j++) {
                if (s.charAt(j) == pre) ++count;
                else {
                    builder.append(count);
                    builder.append(pre);
                    pre = s.charAt(j);
                    count = 1;
                }
            }
            builder.append(count);
            builder.append(pre);

            s = builder.toString();
            builder.delete(0, builder.length());
//            System.out.println(i + " --> " + s);
        }

        return s;
```

```
        }
    }
```

implement-strstr

```
package algorithm.string;

/**
 * https://leetcode.com/problems/implement-strstr/
 *
 Implement strStr().

 Returns the index of the first occurrence of needle in haystack,
 or -1 if needle is not part of haystack.
 *
 * @author xiaobaoqiu  Date: 16-7-11 Time:  下午10:26
 */
public class ImplementStrStr {
    public static void main(String[] args) {
        String[][] cases = new String[][] {
            new String[]{"abcd", "ab"},
            new String[]{"abcd", "bc"},
            new String[]{"abcd", "cd"},
            new String[]{"abcd", "abcd"},
            new String[]{"abcd", "abcde"},
            new String[]{"abcd", ""},
            new String[]{"abcd", "abce"},
        };

        for (String[] v : cases) {
            System.out.println(v[0] + "," + v[1] + " --> " + strStr(v[0], v
        }
    }

    /**
     * 暴力
     * 时间:O(m*n)
     * 空间:O(1)
     *
     * 2 ms
     * Your runtime beats 66.79% of java submissions
     */
    public static int strStr(String haystack, String needle) {
        if (haystack == null || needle == null) return -1;
        if (haystack.length() < needle.length()) return -1;
        if (haystack.length() == needle.length()) return haystack.equals(ne

        for (int i = 0; i < haystack.length() - needle.length() + 1; ) {
            int j = 0;
            for (; j < needle.length(); j++) {
                if (haystack.charAt(i + j) != needle.charAt(j)) break;
            }
            if (j == needle.length()) return i;
            i++;
        }
        return -1;
    }

    /**
```

```
     * KMP
     * TODO
     */
    public static int strStr_1(String haystack, String needle) {
        return 0;
    }
}
```

isomorphic-strings

```
package algorithm.string;

import java.util.HashMap;
import java.util.Map;

/**
 * https://leetcode.com/problems/isomorphic-strings/
 *
 Given two strings s and t, determine if they are isomorphic.

 Two strings are isomorphic if the characters in s can be replaced to get t

 All occurrences of a character must be replaced with another character
 while preserving the order of characters.
 No two characters may map to the same character
 but a character may map to itself.

 For example,
 Given "egg", "add", return true.

 Given "foo", "bar", return false.

 Given "paper", "title", return true.

 * @author xiaobaoqiu  Date: 16-7-4 Time:  下午11:06
 */
public class IsomorphicStrings {
    public static void main(String[] args) {
        System.out.println(isIsomorphic_1("egg", "add"));   //true
        System.out.println(isIsomorphic_1("foo", "bar"));   //false
        System.out.println(isIsomorphic_1("paper", "title"));   //true
        System.out.println(isIsomorphic_1("ab", "aa")); //false
        System.out.println(isIsomorphic_1("ab", "ca")); //true
    }

    /**
     * 34 ms
     * Your runtime beats 26.59% of java submissions
     */
    public static boolean isIsomorphic(String s, String t) {
        if ((s == null && t == null)) return true;
        if ((s == null || t == null)) return false;
        if (s.isEmpty() && t.isEmpty()) return true;
        if (s.length() != t.length()) return false;

        Map<Character, Character> map = new HashMap<Character, Character>()
        for (int i = 0; i < s.length(); i++) {
            if (map.containsKey(s.charAt(i))) {
                if (t.charAt(i) != map.get(s.charAt(i))) return false;
            } else {
                map.put(s.charAt(i), t.charAt(i));
            }
        }
        map.clear();
```

```
        for (int i = 0; i < t.length(); i++) {
            if (map.containsKey(t.charAt(i))) {
                if (s.charAt(i) != map.get(t.charAt(i))) return false;
            } else {
                map.put(t.charAt(i), s.charAt(i));
            }
        }
        return true;
    }

    /**
     * 48 ms
     * Your runtime beats 12.76% of java submissions
     */
    public static boolean isIsomorphic_1(String s, String t) {
        if ((s == null && t == null)) return true;
        if ((s == null || t == null)) return false;
        if (s.isEmpty() && t.isEmpty()) return true;
        if (s.length() != t.length()) return false;

        Map<Character, Character> fMap = new HashMap<Character, Character>(
        Map<Character, Character> bMap = new HashMap<Character, Character>(
        for (int i = 0; i < s.length(); i++) {
            if (fMap.containsKey(s.charAt(i))) {
                if (t.charAt(i) != fMap.get(s.charAt(i))) return false;
            } else {
                fMap.put(s.charAt(i), t.charAt(i));
            }

            if (bMap.containsKey(t.charAt(i))) {
                if (s.charAt(i) != bMap.get(t.charAt(i))) return false;
            } else {
                bMap.put(t.charAt(i), s.charAt(i));
            }
        }
        return true;
    }
}
```

length-of-last-word

```
package algorithm.string;

/**
 * https://leetcode.com/problems/length-of-last-word/
 *
 Given a string s consists of upper/lower-case alphabets
 and empty space characters ' ',
 return the length of last word in the string.

 If the last word does not exist, return 0.

 Note: A word is defined as a character sequence consists of
 non-space characters only.

 For example,
 Given s = "Hello World",
 return 5.

 * @author xiaobaoqiu  Date: 16-7-7 Time:  下午11:28
 */
public class LengthOfLastWord {
    public static void main(String[] args) {
//        String s = "Hello    World";
        String s = "Hello    World    ";
        System.out.println(lengthOfLastWord(s));
        System.out.println(lengthOfLastWord_1(s));
    }

    /**
     * 2 ms
     * Your runtime beats 38.72% of java submissions.
     */
    public static int lengthOfLastWord(String s) {
        if (s == null || s.isEmpty()) return 0;
        String[] array = s.split(" ");
        if (array.length == 0) return 0;
        return array[array.length - 1].length();
    }

    /**
     * 自己扫描
     * 0 ms
     * Your runtime beats 56.23% of java submissions.
     */
    public static int lengthOfLastWord_1(String s) {
        if (s == null || s.isEmpty()) return 0;
        int l = 0;
        for (int i = s.length() - 1; i >= 0; i--) {
            if (s.charAt(i) == ' ') {
                if (l == 0) continue;
                else break;
            }
            l++;
        }
```

```
        return l;
    }
}
```

maximum-product-of-word-lengths

```
package algorithm.string;

import java.util.Arrays;

/**
 * https://leetcode.com/problems/maximum-product-of-word-lengths/
 *
 Given a string array words, find the maximum value of length(word[i]) * le
 where the two words do not share common letters.
 You may assume that each word will contain only lower case letters. If no

 Example 1:
 Given ["abcw", "baz", "foo", "bar", "xtfn", "abcdef"]
 Return 16
 The two words can be "abcw", "xtfn".

 Example 2:
 Given ["a", "ab", "abc", "d", "cd", "bcd", "abcd"]
 Return 4
 The two words can be "ab", "cd".

 Example 3:
 Given ["a", "aa", "aaa", "aaaa"]
 Return 0
 No such pair of words.

 * @author xiaobaoqiu  Date: 16-5-24 Time:  下午9:12
 */
public class MaximumProductOfWordLengths {
    public static void main(String[] args) {
        String[] s = new String[] {"a", "ab", "abc", "d", "cd", "bcd", "abc
        System.out.println(maximumProduct(s));
    }

    /**
     * 125 ms
     * Your runtime beats 14.70% of java submissions
     */
    public static int maximumProduct(String[] words) {
        int[] dict = new int[26];
        int product = 0, iLength = 0;
        for (int i = 0; i < words.length; i++) {
            iLength = words[i].length();
            for (int j = 0; j < 26; j++) dict[j] = 0;   //Arrays.fill(dict,
            for (int j = i + 1; j < words.length; j++) {
                if (words[j].length() * iLength <= product || intersect(wor
                if (words[j].length() * iLength > product) product = words[
            }
        }
        return product;
    }

    private static boolean intersect(String left, String right, int[] dict)
        for (int i = 0; i < left.length(); i++) {
```

```
            dict[left.charAt(i) - 'a'] = 1;
        }
        for (int i = 0; i < right.length(); i++) {
            if (dict[right.charAt(i) - 'a'] == 1) return true;
        }
        return false;
    }
}
```

reverse-string

```
package algorithm.string;

/**
 * https://leetcode.com/problems/reverse-string/

 Write a function that takes a string as input and returns the string rever

 Example:
 Given s = "hello", return "olleh".

 Subscribe to see which companies asked this question

 * @author xiaobaoqiu  Date: 16-5-17 Time:  下午9:03
 */
public class ReverseString {
    public static void main(String[] args) {
        String s = "hello";
        System.out.println(reverseString(s));
    }

    /**
     * 6 ms
     */
    public static String reverseString(String s) {
        if (s == null || s.isEmpty()) return s;

        StringBuilder builder = new StringBuilder(s.length());
        for (int i = s.length() - 1; i >= 0; i-- ) {
            builder.append(s.charAt(i));
        }
        return builder.toString();
    }
}
```

reverse-vowels-of-a-string

```
package algorithm.string;

/**
 * https://leetcode.com/problems/reverse-vowels-of-a-string/
 * <p/>
 * Write a function that takes a string as input and reverse only the vowe
 * <p/>
 * Example 1:
 * Given s = "hello", return "holle".
 * <p/>
 * Example 2:
 * Given s = "leetcode", return "leotcede".
 *
 * @author xiaobaoqiu  Date: 16-5-28 Time:  上午12:31
 */
public class ReverseVowelsOfAString {
    public static void main(String[] args) {
        String i = "bcd";
//        String i = "hello";
//        String i = "leetcode";
        System.out.println(i);
        System.out.println(reverseVowels(i));
    }

    /**
     * 翻转元音字符
     *
     * 5 ms
     */
    public static String reverseVowels(String s) {
        if (s == null || s.length() <= 1) return s;
        char[] arr = s.toCharArray();
        for (int i=0, j=arr.length-1; i<j;) {
            while (!isVowels(arr[i]) && i<j) i++;
            while (!isVowels(arr[j]) && i<j) j--;
            if (i < j) {
                char t = arr[i];arr[i] = arr[j];arr[j]=t;
                i++;j--;
            }
        }

        return String.valueOf(arr);
    }

    private static boolean isVowels(char c) {
        return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
        c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U';
    }
}
```

roman-to-integer

```
package algorithm.string;

import java.util.HashMap;
import java.util.Map;

/**
 * https://leetcode.com/problems/roman-to-integer/
 *
 Given a roman numeral, convert it to an integer.

 Input is guaranteed to be within the range from 1 to 3999.

 * @author xiaobaoqiu  Date: 16-5-25 Time:  下午9:29
 */
public class RomanToInteger {
    /**
     *
     *  | (1) 、X (10) 、C (100) 、M (1000) 、V (5) 、L (50) 、D (500)
     *
     char* c[4][10]={
     {"","I","II","III","IV","V","VI","VII","VIII","IX"}, 1->9
     {"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"}, 10->90
     {"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"}, 100->900
     {"","M","MM","MMM"} 1000->3000
     };
     string roman;
     roman.append(c[3][num / 1000 % 10]);
     roman.append(c[2][num / 100 % 10]);
     roman.append(c[1][num / 10 % 10]);
     roman.append(c[0][num % 10]);
     * @param args
     */
    public static void main(String[] args) {
//         | (1) 、X (10) 、C (100) 、M (1000) 、V (5) 、L (50) 、D (500)
        System.out.println('A' + " --> " + (int)('A'));
        System.out.println('I' + " --> " + (int)('I'));
        System.out.println('X' + " --> " + (int)('X'));
        System.out.println('C' + " --> " + (int)('C'));
        System.out.println('M' + " --> " + (int)('M'));
        System.out.println('V' + " --> " + (int)('V'));
        System.out.println('L' + " --> " + (int)('L'));
        System.out.println('D' + " --> " + (int)('D'));
        String[] test = new String[]{
                "I", "X", "C", "M", "V", "L", "D"
//                "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"  //1000->3000
        };

        for (String s : test) {
            System.out.println(s + " --> " + romanToInt(s));
        }
```

```
        //"MDCCCLXXXIV"
    }

    /**
     * 规则: https://www.douban.com/note/335254352/
     *
     * 7 ms
     * Your runtime beats 82.35% of java submissions
     */
    public static int romanToInt(String s) {
        int[] map = new int[26];    //A --> 65
        map[8] = 1;         //I --> 73
        map[21] = 5;        //V --> 86
        map[23] = 10;       //X --> 88
        map[11] = 50;       //L --> 76
        map[2] = 100;       //C --> 67
        map[3] = 500;       //D --> 68
        map[12] = 1000;     //M --> 77
        if (s.length() == 1)  return map[s.charAt(0) - 65];

        int preValue = map[s.charAt(0) - 'A'];
        int sum = preValue, curValue = 0;
        for (int i = 1; i < s.length(); i++) {
            curValue = map[s.charAt(i) - 'A'];
            if (preValue < curValue) {
                sum = sum - 2 * preValue + curValue;
            } else {
                sum += curValue;
            }
            preValue = curValue;
        }
        return sum;
    }
}
```

valid-anagram

```
package algorithm.string;

import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

/**
 * https://leetcode.com/problems/valid-anagram/
 *
 Given two strings s and t, write a function to determine if t is an anagra

 For example,
 s = "anagram", t = "nagaram", return true.
 s = "rat", t = "car", return false.

 Note:
 You may assume the string contains only lowercase alphabets.

 * @author xiaobaoqiu  Date: 16-5-22 Time:  下午5:13
 */
public class ValidAnagram {
    public static void main(String[] args) {
        //题目意思：字母出现次数相同
        String s = "anagram";
        String t = "nagaram";

//        String s = "rat";
//        String t = "car";

        System.out.println(isAnagram_1(s, t));
    }

    /**
     * 45 ms
     * Your runtime beats 20.90% of java submissions
     */
    public static boolean isAnagram(String s, String t) {
        Map<Character, Integer> sCount = new HashMap<Character, Integer>()
        for (int i = 0; i<s.length(); i++) {
            Character c = s.charAt(i);
            if (!sCount.containsKey(c)) sCount.put(c, 1);
            else sCount.put(c, sCount.get(c) + 1);
        }

        Map<Character, Integer> tCount = new HashMap<Character, Integer>()
        for (int i = 0; i<t.length(); i++) {
            Character c = t.charAt(i);
            if (!tCount.containsKey(c)) tCount.put(c, 1);
            else tCount.put(c, tCount.get(c) + 1);
        }

        for (Map.Entry<Character, Integer> e : sCount.entrySet()) {
            if (!e.getValue().equals(tCount.get(e.getKey()))) return false
        }
```

```
        for (Map.Entry<Character, Integer> e : tCount.entrySet()) {
            if (!e.getValue().equals(sCount.get(e.getKey()))) return false;
        }
        return true;
    }

    /**
     * 排序
     *
     * 8 ms
     * Your runtime beats 46.36% of java submissions
     */
    public static boolean isAnagram_1(String s, String t) {
        if (s.length() != t.length()) return false;

        char[] sArray = s.toCharArray();
        char[] tArray = t.toCharArray();
        Arrays.sort(sArray);
        Arrays.sort(tArray);
        for (int i = 0; i < s.length(); i++) {
            if (sArray[i] != tArray[i]) return false;
        }

        return true;
    }
}
```