bulls-and-cows

```
package algorithm.hash;

import java.util.HashSet;
import java.util.Set;

/**
 * https://leetcode.com/problems/bulls-and-cows/
 *
 You are playing the following Bulls and Cows game with your friend: You wr

 For example:

 Secret number:  "1807"
 Friend's guess: "7810"
 Hint: 1 bull and 3 cows. (The bull is 8, the cows are 0, 1 and 7.)
 Write a function to return a hint according to the secret number and frien

 Please note that both secret number and friend's guess may contain duplica

 Secret number:  "1123"
 Friend's guess: "0111"
 In this case, the 1st 1 in friend's guess is a bull, the 2nd or 3rd 1 is a
 You may assume that the secret number and your friend's guess only contain

 * @author xiaobaoqiu  Date: 16-7-1 Time:  下午10:42
 */
public class BullsAndCows {
    public static void main(String[] args) {
        System.out.println(getHint("1807", "7810"));    //1A3B
        System.out.println(getHint("1123", "0111"));    //1A1B
        System.out.println(getHint("11", "11"));        //2A0B
        System.out.println(getHint("1122", "2211"));    //0A4B
    }

    /**
     * 8 ms
     * Your runtime beats 39.83% of java submissions
     */
    public static String getHint(String secret, String guess) {
        int A = 0, B = 0;
        int[] dict = new int[10];
        Set<Integer> sameIndex = new HashSet<Integer>();
        for (int i = 0; i < secret.length(); i++) {
            if (secret.charAt(i) == guess.charAt(i)) {
                sameIndex.add(i);
                A++;
            } else {
                dict[secret.charAt(i) - '0'] += 1;
            }
        }
        for (int i = 0; i < guess.length(); i++) {
            if (sameIndex.contains(i)) continue;
            if (dict[guess.charAt(i) - '0'] > 0) {
                B++;
                dict[guess.charAt(i) - '0'] -= 1;
            }
        }
        return A + "A" + B + "B";
    }
}
```

word-pattern

```
package algorithm.hash;

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

/**
 * https://leetcode.com/problems/word-pattern/
 *
 Given a pattern and a string str, find if str follows the same pattern.

 Here follow means a full match, such that there is a bijection
 between a letter in pattern and a non-empty word in str.

 Examples:
 pattern = "abba", str = "dog cat cat dog" should return true.
 pattern = "abba", str = "dog cat cat fish" should return false.
 pattern = "aaaa", str = "dog cat cat dog" should return false.
 pattern = "abba", str = "dog dog dog dog" should return false.
 Notes:
 You may assume pattern contains only lowercase letters,
 and str contains lowercase letters separated by a single space

 * @author xiaobaoqiu  Date: 16-7-7 Time:  下午10:56
 */
public class WordPattern {
    public static void main(String[] args) {
        System.out.println(wordPattern("abba", "dog cat cat dog")); //true
        System.out.println(wordPattern("abba", "dog cat cat fish"));//false
        System.out.println(wordPattern("aaaa", "dog cat cat dog")); //false
        System.out.println(wordPattern("abba", "dog dog dog dog")); //false
    }

    /**
     * 2 ms
     * Your runtime beats 85.56% of java submissions.
     */
    public static boolean wordPattern(String pattern, String str) {
        String[] array = str.split(" ");
        if (pattern.length() != array.length) return false;

        int[] fMap = new int[26];
        Map<String, Integer> bMap = new HashMap<String, Integer>();
        Arrays.fill(fMap, -1);
        int pos;
        for (int i = 0; i < array.length; i++) {
            pos = pattern.charAt(i) - 'a';
            if (fMap[pos] == -1) fMap[pos] = i;
            else {
                if (!array[i].equals(array[fMap[pos]])) return false;
            }

            if (!bMap.containsKey(array[i])) bMap.put(array[i], pos);
            else {
                if (pos != bMap.get(array[i])) return false;
            }
        }

        return true;
    }
}
```