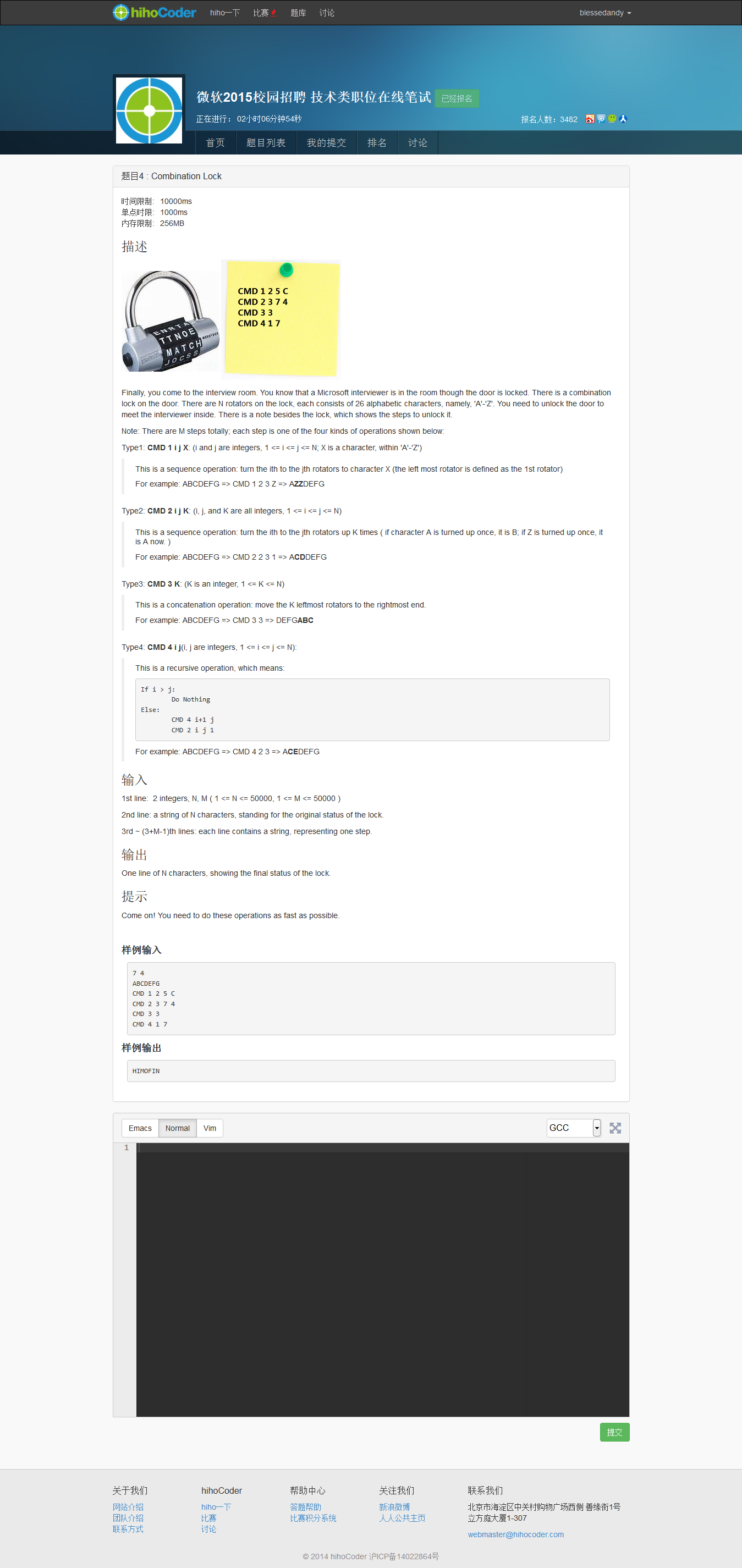
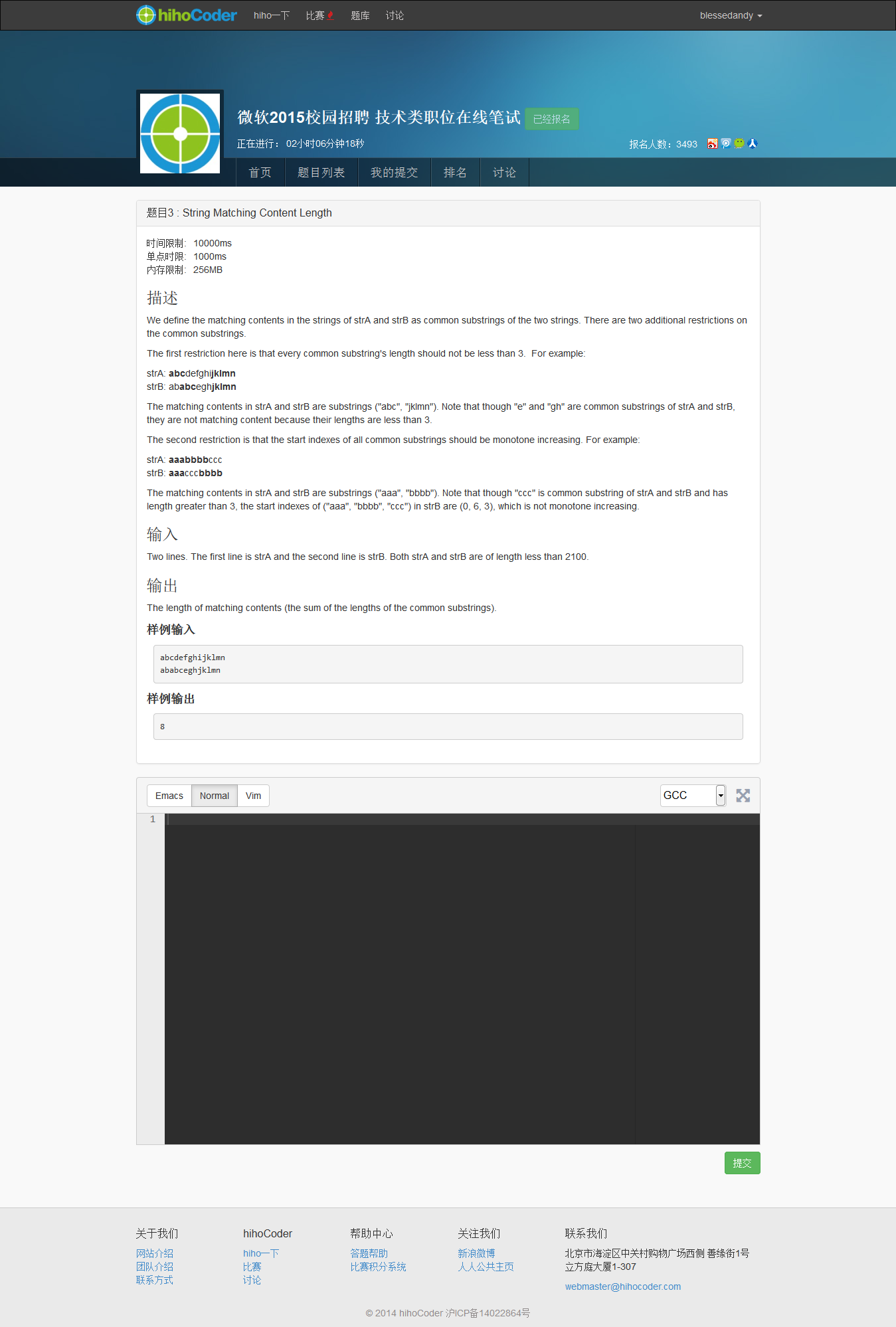
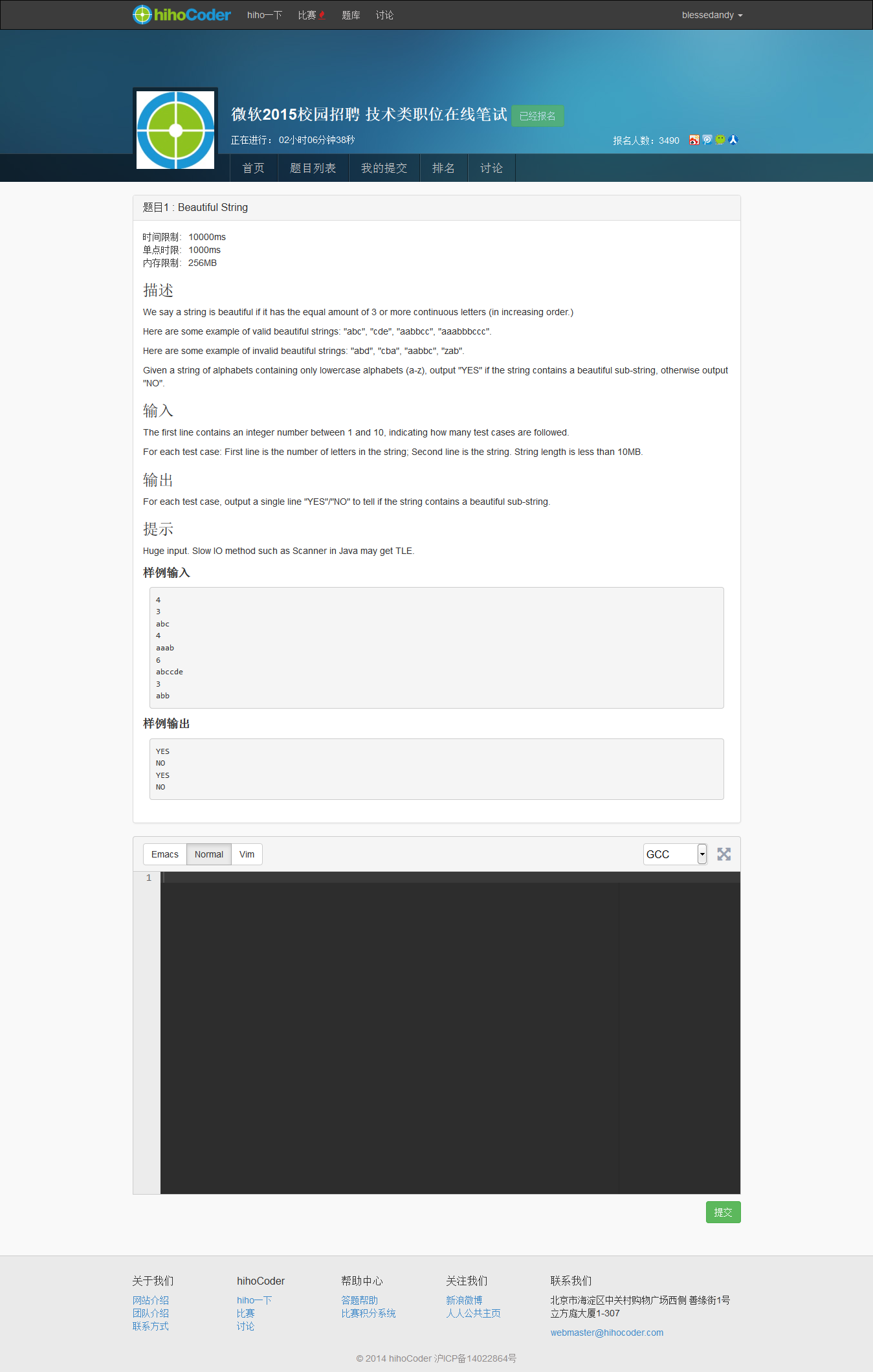
**题目4 : Combination Lock**

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

**描述**

Finally, you come to the interview room. You know that a Microsoft interviewer is in the room though the door is locked. There is a combination lock on the door. There are N rotators on the lock, each consists of 26 alphabetic characters, namely, 'A'-'Z'. You need to unlock the door to meet the interviewer inside. There is a note besides the lock, which shows the steps to unlock it.

Note: There are M steps totally; each step is one of the four kinds of operations shown below:

Type1: **CMD 1 i j X**: (i and j are integers, 1 <= i <= j <= N; X is a character, within 'A'-'Z')

This is a sequence operation: turn the ith to the jth rotators to character X (the left most rotator is defined as the 1st rotator)

For example: ABCDEFG => CMD 1 2 3 Z => A**ZZ**DEFG

Type2: **CMD 2 i j K**: (i, j, and K are all integers, 1 <= i <= j <= N)

This is a sequence operation: turn the ith to the jth rotators up K times ( if character A is turned up once, it is B; if Z is turned up once, it is A now. )

For example: ABCDEFG => CMD 2 2 3 1 => A**CD**DEFG

Type3: **CMD 3 K**: (K is an integer, 1 <= K <= N)

This is a concatenation operation: move the K leftmost rotators to the rightmost end.

For example: ABCDEFG => CMD 3 3 => DEFG**ABC**

Type4: **CMD 4 i j**(i, j are integers, 1 <= i <= j <= N):

This is a recursive operation, which means:

If i > j:

Do Nothing

Else:

CMD 4 i+1 j

CMD 2 i j 1

For example: ABCDEFG => CMD 4 2 3 => A**CE**DEFG

**输入**

1st line:  2 integers, N, M ( 1 <= N <= 50000, 1 <= M <= 50000 )

2nd line: a string of N characters, standing for the original status of the lock.

3rd ~ (3+M-1)th lines: each line contains a string, representing one step.

**输出**

One line of N characters, showing the final status of the lock.

**提示**

Come on! You need to do these operations as fast as possible.

样例输入

7 4

ABCDEFG

CMD 1 2 5 C

CMD 2 3 7 4

CMD 3 3

CMD 4 1 7

样例输出

HIMOFIN