

# 2020 Wuhan University Collegiate Programming Contest (Preliminary Contest)

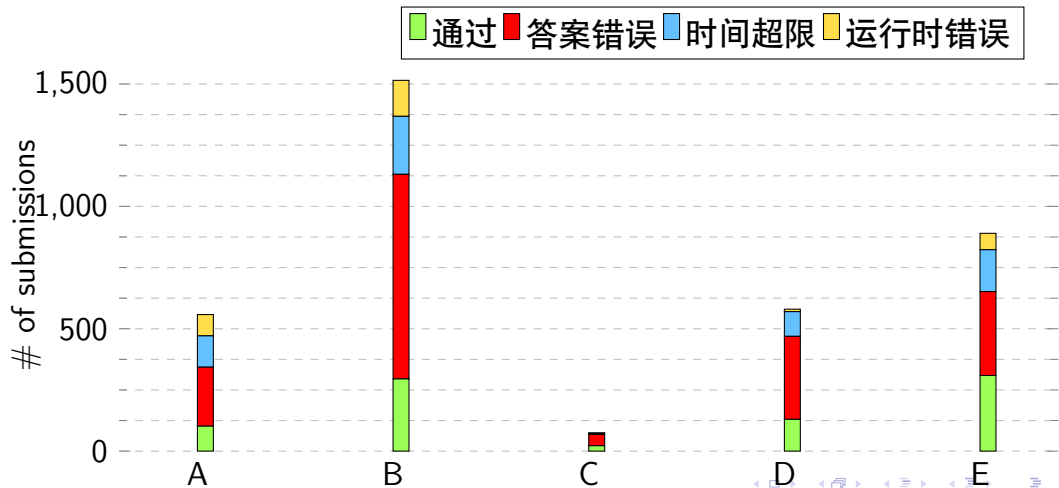
Solutions

Apr. 12th, 2020

# 比赛小结

- 本次比赛共收到 4 122 份提交代码。
- 422 名参赛选手有提交记录。
- 312 名参赛选手至少通过一题。

# 各题通过情况



# E. E Yu is a Brutal Creature

## 题意

找出 0 到  $n$  之间所有满足  $(n+1)|(n^2+1)$  的自然数。

## 题意

找出 0 到  $n$  之间所有满足  $(n+1)|(n^2+1)$  的自然数。

根据平方差公式:  $n^2 - 1 = (n+1)(n-1)$  可知  $(n+1)|(n^2 - 1)$ 。

于是可知  $(n^2 + 1) - (n^2 - 1) = 2$  也应当是  $n+1$  的倍数。

符合条件的数, 只有 1 和 0。

# E. E Yu is a Brutal Creature

## 可能存在的盲点

- 由于它位于最后一个题，所以没有及时发现这是个签到题。
- 这题由于审题不够仔细，一开始写错了数据范围（这一点深感抱歉），在  $N = 10^9$  的情况下暴力枚举显然是行不通的。

## 花絮

这题并不是题面中出现的人鼠出的题。

这题是我们原本设想的最送温暖的题目，但是实际上它的通过数没有 B 题多，着实让我们有点意外。

## B. Best Match

### 题意

给定一个数组，求有多少个数对  $a_i, a_j, i \neq j$  满足  $a_i + a_j == 0$ 。

## B. Best Match

### 题意

给定一个数组，求有多少个数对  $a_i, a_j, i \neq j$  满足  $a_i + a_j == 0$ 。

我们需要记录下数组中每个权值的出现次数，记数组中权值  $i$  的出现次数为  $cnt_i$ 。那么答案就是：

$$\sum_{i=1}^{\max\{a\}} cnt_i \times cnt_{-i} + cnt_0 \times (cnt_0 - 1) / 2$$

算出  $cnt$  数组，直接统计即可。



## B. Best Match

### 可能存在的盲点

- 对于 C/C++，这一题的答案超过了 int 范围，你可能需要使用 long long。
- 这一题主要是很多人都没有搞清楚题目到底需要什么，但考虑到最终大部分同学还算通过了此题，所以也没做补充说明。

### 花絮

这题也不是题面中出现的人鼠出的题。

# A. A Monument for Heroes

## 题意

给你若干字符串，求按照首尾字母相同的方式接龙能接上多少个，且必须按照题目输入的顺序接，也就是先出现的字符串必须接在前面。

# A. A Monument for Heroes

## 题意

给你若干字符串，求按照首尾字母相同的方式接龙能接上多少个，且必须按照题目输入的顺序接，也就是先出现的字符串必须接在前面。

可以通过 DP 实现，记  $dp[i][j]$  表示以  $i$  开头， $j$  结尾的接龙的最长长度。

# A. A Monument for Heroes

## 题意

给你若干字符串，求按照首尾字母相同的方式接龙能接上多少个，且必须按照题目输入的顺序接，也就是先出现的字符串必须接在前面。

可以通过 DP 实现，记  $dp[i][j]$  表示以  $i$  开头， $j$  结尾的接龙的最长长度。随后依次枚举每个字符串，假设字符串  $s$  的开头是  $c_1$ ，结尾为  $c_2$ ，那么更新所有  $dp[i][c_2]$ ，更新方式为  $dp[i][c_2] = \max\{dp[i][c_2], dp[i][c_1] + |s|\}$ 。

# A. A Monument for Heroes

## 可能存在的盲点

- 对于只有一个串该如何处理？
- 暴力搜索类的算法复杂度都不怎么靠谱。

## 花絮

这题一开始出题人没有加按顺序接这个限制，然后验题的时候发现不太对劲，为了避免让出题人再费心想一道题，提出了加限制顺序这个方法抢救了出题人的这个题。正好，原定这里需要一道 Div2 B 难度的动态规划。然而验题人只是简单的加了一段表示需要严格顺序。并没有想到原文中还会有顺序不限的表述，因此没有修改它，对大家题意理解造成的不便深感歉意。

### 题意

给你一个由大写字母构成的二维矩阵，你需要找到一个最大的正方形，使得这个正方形内只包含一种字母。

### 题意

给你一个由大写字母构成的二维矩阵，你需要找到一个最大的正方形，使得这个正方形内只包含一种字母。

本题实际上有多种通过方法，下面介绍两种参考方法：

## D. DIY Masks at Home

### 解法一：暴力哈希

将原矩阵内每一种字母都替换成一个素数，然后计算这个矩阵的二维前缀积（对大素数取模）。那么在给定二分长度  $k$  的前提下，我们每次可以枚举一个矩形的左上角  $(i, j)$ ，利用逆元计算出这么个矩形的积，再和这种字母的纯  $k * k$  正方形对应的哈希值比对。

如果担心碰撞，只需改成双哈希就好。总复杂度为常数有点大的  $O(nm \log(n))$ 。



## D. DIY Masks at Home

### 解法二

我们如果在原矩阵  $F$  的基础上预处理出一个新矩阵  $G$ , 第  $i$  行第  $j$  列的值的意义为: 这个值在这一行前面有多少个连续的数和它相同 (包括自己)。随后我们对于每一列从上到下遍历, 如果一个边长为  $k$ , 右下角位于  $(i, j)$  的矩形存在的话, 一定会满足:  $\min_{i-k < t \leq i} \{G[t][j]\} \geq k$

### 解法二

可想而知，对于我们枚举的右下角，右上角也是具备单调性的，所以我们可以采用二分 + 对每一列维护 RMQ 数组的方法获得一个  $O(nm \log(n))$  的方法。

### 解法二

可想而知，对于我们枚举的右下角，右上角也是具备单调性的，所以我们可以采用二分 + 对每一列维护 RMQ 数组的方法获得一个  $O(nm \log(n))$  的方法。

Bonus: 实际上这题还存在  $O(nm)$  的做法，此处不做赘述，留给各位观众自行思考吧。

## D. DIY Masks at Home

### 可能存在的盲点

按照原题目的题意，一个字母仅代表这么一个小方块的颜色，我只需要切下一个正方形就好了。

那么……一定必须按照横平竖直的方法切吗……？

## D. DIY Masks at Home

### 可能存在的盲点

按照原题目的题意，一个字母仅代表这么一个小方块的颜色，我只需要切下一个正方形就好了。

那么……一定必须按照横平竖直的方法切吗……？

### 花絮

这题原出题人使用的是解法一，后来实际验题的时候发现这玩意常数实在是大，在牛客上硬生生跑了 4s，为了鼓励大家想想这种做法以及其它的想法，此题才采用了看上去比较吓人的时限。

后来验题人想到了解法二正经的复杂度的做法，以及更进一步的做法，于是后来干脆就用  $O(nm)$  的方法造了数据。

# C. Can You Help ZSGW

## 题意

有一个排列，已知我们对于这个排列执行单调栈算法过程中，遍历到每一个位置之后单调栈的大小，有些位置缺失可以任意。求一个满足这种情况的字典序最小的排列。

## C. Can You Help ZSGW

### 解法

首先我们应该做的事，是补全这个单调栈数组  $b$ 。单调栈数组由于单调栈算法的特点，必然满足这么几个特点：

1.  $b[1] = 1$
2. 若  $b[i] > b[i-1]$ ，则  $b[i] = b[i-1] + 1$ ，且  $a[i] > a[i-1]$ 。
3. 对于  $b[i] \leq b[i-1]$ ，一定有  $a[i] < a[i-1]$ 。

## C. Can You Help ZSGW

### 解法

我们从左到右依次补全每一个为  $-1$  的格子，那么策略应该是：

1. 若  $i = 1$ ,  $b[i] = 1$ 。
2. 否则，如果我们填一个比  $b[i - 1]$  小的数的话，就意味着之后补全的时候  $a[i - 1] > a[i]$ ，在字典序上不会是个好主意。所以应该填  $b[i - 1] + 1$ 。



## C. Can You Help ZSGW

### 解法

补全数组之后，规律如下：

首先我们可以发现所有为 1 的位置组成了以 1 截止的降序序列。然后对于每个被 1 分割的子区间，2 也会满足类似的规律。然后对于 2 进一步分割的子区间 3 也会如此。

## C. Can You Help ZSGW

### 解法

补全数组之后，规律如下：

首先我们可以发现所有为 1 的位置组成了以 1 截止的降序序列。然后对于每个被 1 分割的子区间，2 也会满足类似的规律。然后对于 2 进一步分割的子区间 3 也会如此。

于是我们按照权值递增的顺序，依次填入每个数，然后分治递归填写子区间。标程的复杂度为  $O(n \log(n))$ ，实际上利用分治区间的单调性可以进一步优化到  $O(n)$ 。

## C. Can You Help ZSGW

### 花絮

文中出现的人名 ZSGW 实则是出题人高中的反写。

本题源自于作者在参加某次集训时，一不小心读错读成的一个题。由于现场这个题对拍死活不出问题，让作者印象深刻，同时也让作者敢把自己的方法确定为正确做法。

本位置原本不使用这个题，原题因为一位验题人验了一个周六表示没写出来而被撤下来，换成了这个相对好写点的题。

### 花絮 2

本题居然在一个月之前被原封不动地出过一次。

题号是洛谷 P6198，真的原封不动。

**声明：**本题是我今年 1 月参加某集训时由于读错题而进行改编的一道新题，此题的想法在今年 1 月，也就是洛谷该题出现之前（3 月月赛）即已形成，绝不是抄袭或者偷题。

据分析，有可能是这位出题人也参加过这个集训吧。

## C. Can You Help ZSGW

### 花絮 2

我本人是非常在意出题出原题这种事情的（当然新生赛出现了一道 CF 原题，那是实在因为时间紧迫迫不得已），无论如何出原题都是对比赛体验的一种破坏，因此对于今天发生的这么一件事情深感歉意。

# 总结

本次线上赛存在许多小瑕疵，主要原因应该还是我们出题人。由于疫情影响，我们出题组的一切交流只能依靠线上，极大地降低了交流效率。再次对于题面中可能出现的影响比赛体验的地方表示歉意。

题目质量本身的话，也没有特别高质量的一道题，唯一一道我觉得还挺不错的题最后还被挂说是原题。

我们会努力让正赛的体验更好的。