

# 2022牛客寒假算法基础集训营 第 1 场

出题人: fried-chicken



# 预估难度

思维 知识 实现 综合

F	81	84	90	85
G	86	72	85	81
K	74	71	83	76
B	66	72	75	71
D	75	80	55	70
I	83	70	48	67
F	91	55	40	62
H	65	62	53	60
A	55	65	60	60
C	43	45	68	52
J	45	50	52	49
E	56	40	42	46
L	30	35	37	34

- **思维**：指题目对思考的要求，具体来说，包括题目做法的灵感难度、思考难度、思考深度等方面；
- **知识**：指作出该题目所需知识的要求，包括但不限于题目对算法、数据结构、常见trick的要求，评价标准为对知识广度、深度、难度的要求；
- **实现**：指实现该题算法的代码难度，包括对实现技巧的要求、对代码量的要求、对细节处理的要求等。
- 题目特点：整体难度不高；不同出题人有不同的出题风格，这场题目风格比较偏思维，知识点和实现难度较低。（广告：如果你觉得做这种题目风格有困难，可以补补我的小白月赛38的题233）
- 四种颜色由下到上代表了本场的四种难度：
  - easy, mid-easy, mid, mid-hard
  - （为甚么没有hard？是我难度评估太保守了吗？）
  - （在接下来的寒假集训营中，你会知道什么才是hard~~~）

# 过题情况 (不同时间)

[比赛说明](#)[题目](#)[提交](#)[排名](#)[管理比赛](#)

距离比赛结束还有: 04 时 29 分 52 秒

☐ 查看我关注的人

排行榜将在 2022-01-24 18:00:00 比赛结束时封榜，比赛时间段之外的提交不计入排行。 [点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					24 60	1 284	1 8	2 16	555 2638	17 23	0 0	39 187	2 2	18 54	2 2	1666 2142

[比赛说明](#)[题目](#)[提交](#)[排名](#)[管理比赛](#)

距离比赛结束还有: 03 时 59 分 39 秒

☐ 查看我关注的人

排行榜将在 2022-01-24 18:00:00 比赛结束时封榜，比赛时间段之外的提交不计入排行。 [点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					58 171	2 536	5 23	10 36	1248 6567	55 76	0 0	181 900	4 7	158 412	5 8	2269 3135

# 过题情况 (不同时间)

比赛说明

题目

提交

排名

管理比赛

距离比赛结束还有: 02 时 59 分 57 秒

Q

输入用户名称/学校名称回车

☐ 查看我关注的人

排行榜将在 2022-01-24 18:00:00 比赛结束时封榜，比赛时间段之外的提交不计入排行。[点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					155 427	6 864	23 84	48 198	1838 10626	229 314	2 4	503 2725	18 47	838 2674	13 27	2705 4001

比赛说明

题目

提交

排名

管理比赛

距离比赛结束还有: 01 时 59 分 07 秒

Q

输入用户名称/学校名称回车

CD多起来了，大家注意到CD其实不难了

☐

查看我关注的人

排行榜将在 2022-01-24 18:00:00 比赛结束时封榜，比赛时间段之外的提交不计入排行。[点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					265 747	17 1006	101 296	165 544	2119 12565	546 808	4 18	723 4138	72 188	1348 5017	20 55	2887 4449

# 过题情况 (不同时间)

比赛说明 题目 提交 排名 管理比赛

距离比赛结束还有: 00 时 59 分 37 秒

☐ 查看我关注的人

排行榜将在 2022-01-24 18:00:00 比赛结束时封榜，比赛时间段之外的提交不计入排行。 [点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					368 1024	40 1158	242 883	326 1052	2279 13713	823 1331	12 60	854 5146	146 421	1591 6398	41 118	2987 4708

比赛说明 题目 提交 排名 解题&讨论 NEW 管理比赛

比赛已结束，去参与讨论

榜单范围设置 ☒ 查看我关注的人

排行榜已在 2022-01-24 18:00:00 比赛结束后正式封榜，比赛时间段之外的提交不计入排行。 [点击查看计分规则。](#)

名次	参赛者	学校	通过	罚时	A	B	C	D	E	F	G	H	I	J	K	L
					428 1273	72 1379	402 1701	451 1636	2393 14621	1007 1695	19 167	966 6060	250 792	1728 7288	68 228	3106 4953

# 花絮



我们真能猜



您们真能问



官方真能蹭  
(虽然其实是我发的)

# 过题情况 (一血)

L: 02:10 by [zucc\\_酒杯](#)  
E: 03:21 by [fuzhiji](#)  
F: 04:46 by [uryuuu](#)  
H: 06:47 by [Dewset](#)  
J: 06:58 by [LeiLeiKunLe](#)  
A: 08:46 by [ZAFU\\_杨成艺](#)  
C: 11:37 by [w34](#)  
D: 13:24 by [Z0136](#)  
B: 23:43 by [Huah](#)  
I: 27:40 by [shjzhqm](#)  
K: 27:46 by [uryuuu](#)  
G: 73:17 by [shjzhqm](#)  
AK: 139:23 by [牛客780183216号](#)

# L. 牛牛学走路

- 恭喜你，签到成功！



## E. 炸鸡块君的高中回忆

- 注意 $n=m=1$ 的特判;
- 模拟会超时, 要推出公式;
- 公式不唯一, 能过的公式都是好公式。

```
if(m==1){
    if(n==1)    puts("1");
    else       puts("-1");
}
else{
    printf("%d\n", ((n-1)/(m-1) + ((n-1)%(m-1) != 0)) * 2 - 1);
}
```

## J. 小朋友做游戏

- 先将两种小朋友的幸福度分别按从大到小排序，记为A和B数组；
- 那么最优的方案一定是从A和B中各选一个前缀；
- 因此可以求出两个数组的前缀和，然后枚举从A中选了多少人（从B中选的人数等于总人数减去A中的），利用前缀和 $O(1)$ 的获得此时的总幸福度；
- 对于圆圈紧挨着的限制，其实就相当于限制闹腾小朋友最多选 $\frac{n}{2}$ 个，在上面枚举的时候加入该限制即可；

# C.Baby's first attempt on CPU

- 本题定位大概是一道题面偏实际应用、需要仔细阅读理解题面、代码量不是很大但需要多加思考且不好写的题目，个人感觉是个质量挺高的小模拟题（自满中）；
- 本题主要问题可能就是，选手们想太多寄存器的细节了，于是提问区有好多各种各样关于寄存器的问題，但其实其实寄存器这个东西可以抽象掉，**输入可以完全等价于要求某些语句之间最少隔三句**，然后插入就完事儿了；
- 做法：直接模拟这个过程，开一个数组B存储插入后的结果。遍历输入的原数组A，看B最后的几个语句是否有和当前语句冲突的，若有，则在B末尾加适当空语句。
- 注意B要开的够大。
- 花絮：也是出题过程中被改的次数（出锅+改题面）最多的题。

# A. 九小时九个人九扇门

- 有用的结论：一个数的数字根等于这个数对9取模的结果（特别地，取模得0则数字根为9）；
- ↑可以通过证明每一轮迭代不改变数字对9取模的结果来证明该结论；
- 问题转化为：从 $\{a_n\}$ 中选择一些数字使得其求和对9取模得0,1,2,3,4,5,6,7,8有多少种选法；
- 这是一个经典的0/1背包变形的DP问题；
- 令 $dp[i][j]$ 表示考虑了前 $i$ 个数，选择了一些数字使得求和对9取模得 $j$ 的方案数；
- 转移类似0/1背包的转移。

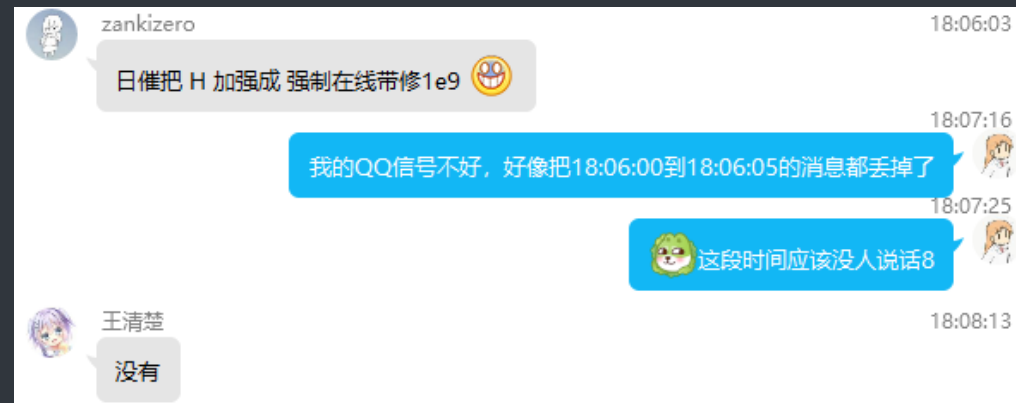
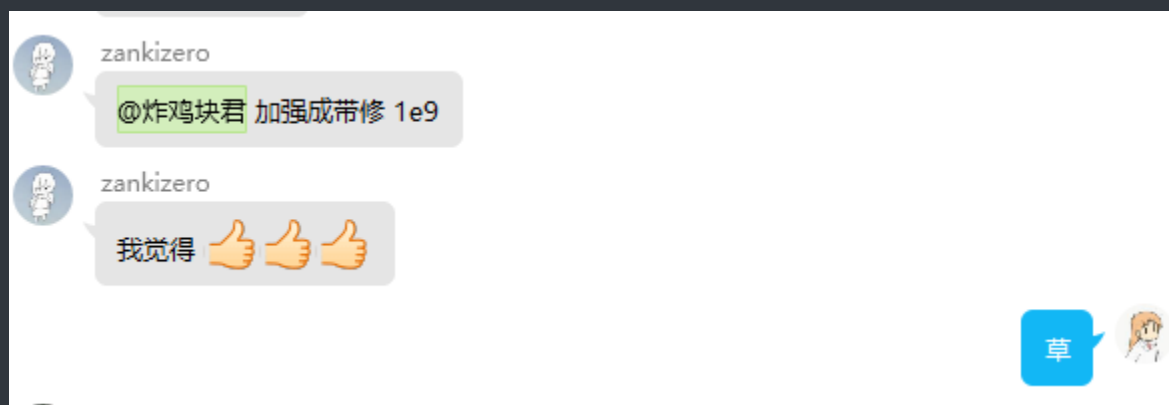
# H. 牛牛看云

- 本题没做出来的xdm要提高对数据范围的敏感程度~
- 注意到 $n \leq 10^6$ 而 $a[i] \leq 1000$ ，可以看到 $a[i]$ 范围很小，或者说极限数据下会有大量重复的值出现，我们想想怎么利用这一点；
- 记 $\text{cnt}[i]$ 表示 $i$ 出现的次数，枚举 $(i, j)$ 对儿（共 $10^6$ 种）；
- 不同情况直接相乘；
- 特殊处理相同情况；

```
ll ans=0;
rep(i,0,1000){
    rep(j,i,1000){
        ll add;
        if(i==j) add=(cnt[i]+cnt[i]*(cnt[i]-1))/2;
        else add=cnt[i]*cnt[j];
        ans=ans+add*(ll)abs(i+j-1000);
    }
}
printf("%lld\n",ans);
```

# H. 牛牛看云

- Bonus: 假设 $a[i]$ 范围可以到 $10^6$ 要怎么做?  $10^9$ 呢? 如果带修改呢;
- 有验题人想要我改成这个版本, 被我正义的拒绝了 (其实也是我懒)



## F. 中位数切分

- 这是一道十分有趣的题目，预期的简单解法 $O(n)$ 且一分钟就可以写完，复杂解法 $O(n\log n)$ 且要写一段时间；
- 简单解法：
- 记数列中 $\geq m$ 的数字有 $cnt_1$ 个， $< m$ 的数字有 $cnt_2$ 个，则答案为 $cnt_1 - cnt_2$ ，该值 $\leq 0$ 时输出-1，几行就写完啦~；
- 复杂解法：
- 使用树状数组+前缀和+离散化+DP，将 $O(n^2)$ 的DP优化到 $O(n\log n)$ ；

## F. 中位数切分

- 证明一（严谨）：记 $f(l, r)$ 为原数组中 $a[l] \dots a[r]$ 一段中的元素对应的 $cnt_1 - cnt_2$ 的值；
- $f()$ 的性质：
  - $f(l, r) > 0$ 表示该段单独拿出来满足中位数 $\geq m$ ；
  - $f(l, r) = f(l, mid) + f(mid + 1, r)$
- 原问题 $f(1, n) \leq 0$ 时输出 $-1$ 是显然的；
- 欲证明： $f(1, n) > 0$ 时， $f(1, n)$ 即为原问题答案；
- 考虑初始所有数组成了一个大区间，我们在通过切大区间为若干段来得到最终答案；
- 若可以找到一个位置 $mid$ ，使 $f(1, mid) > 0 \&\& f(mid + 1, n) > 0$ ，则沿 $mid$ 将数组切开得到的两部分中位数依然满足条件，此时区间数 $+=1$ ；
- 所以我们要探究下什么时候数组可以切；



## F. 中位数切分

- 证明一（严谨）
- 定理：当且仅当 $f(l, r) > 1$ 时存在一种切法使
$$f(l, mid) > 0 \&\& f(mid + 1, r) > 0$$
- 证明：
- 若有一个位置 $mid$ 使得 $f(l, mid) = 1$ ，则该位置是满足条件的切割位置，因为此时 $f(l, mid) = 1 > 0$ ， $f(mid + 1, r) = f(l, r) - f(l, mid) > 1 - 1 = 0$ ；
- 又因为 $f(l, l - 1) = 0$ （表示空区间）且 $f(l, r) > 1$ 且 $f(l, x) \rightarrow f(l, x + 1)$ 时值只会变化1，因此过程中一定存在某一时刻 $mid$ 使得 $f(l, mid) = 1$ 。

## F. 中位数切分

- 证明一（严谨）
- 由上，只要 $f(l, r) > 1$ 就可以切，而我们又希望切得尽可能多，因此最终状态一定是所有切割得到的段都有 $f(l, r) = 1$ （否则还可以再切）；
- 因此， $\sum_{l_i, r_i} f(l_i, r_i) = \text{最终切成的段数}$ ；
- 又因为初始时有 $\sum_{l_i, r_i} f(l_i, r_i) = f(1, n) = cnt_1 - cnt_2$ ；
- 所以最终切成的段数 $= cnt_1 - cnt_2$ ，证毕。

## F. 中位数切分

- 证明二（不严谨）：先假设数组里只有 $\geq m$ 的数字们（想象有一个只含 $\geq m$ 的数的长为 $cnt_2$ 的数组），他们都自己作为一个区间，则一共有 $cnt_1$ 个区间，且这些区间都是满足中位数 $\geq m$ 的要求的；
- 然后考虑插一个 $< m$ 的数字进去，这个数字需要和两个 $\geq m$ 的数组组成一个区间才可以满足中位数要求，因此原来两个区间变成了一个区间，区间数减一，然后删掉这个数和某个 $\geq m$ 的数，数组长变为 $cnt_2 - 1$ ；
- （你可能会说这 $cnt_2 - 1$ 里有一个数其实是在刚才新组成的区间里，这没问题吗，但这种新组成的区间并不影响上述操作，所以可以看作一个 $\geq m$ 的数字）
- 以此类推，可以看出每个 $< m$ 数字的作用就是说区间数减一

## F. 中位数切分

- 复杂做法：使用树状数组+前缀和+离散化+DP，将 $O(n^2)$ 的DP优化到 $O(n\log n)$ ；
- 考虑 $dp[i]$ 表示对于前 $i$ 个数字，最后一段以 $a[i]$ 结尾，最多可以分成多少段；则 $O(n^2)$ 转移为
- $dp[i] = \max_j (dp[j] + 1)$  where  $j < i$  &&  $med(a[j + 1] \dots a[i]) \geq m$
- 我们令所有 $\geq m$ 的 $a[i] = 1$ ， $< m$ 的 $a[i] = -1$ ，并对该数组求前缀和数组记为 $S[]$ ；
- 则 $med(a[j + 1] \dots a[i]) \geq m$ 等价于 $S[i] - S[j] > 0$ ；
- 我们可以在前缀和上开一个维护最大值的树状数组，这样每次就可以求所有 $S[i] - S[j] > 0$ 的 $j$ 当中最大的 $dp[j]$ ；
- 由于 $S[]$ 可正可负，因此还要对其做离散化。

# I. B站与各唱各的

- 答案即为：

$$m \times \frac{2^n - 2}{2^n}$$

- 首先注意到句子与句子之间没有办法互相影响，因此答案是一句话的期望乘以 $m$ ；
- 由于无法交流，每个人在唱每句时唯一的策略就是随机以 $p_i$ 的概率决定唱或不唱这一句（对超算就是拿来生成随机数的，超算好憋屈）；
- 于是失败的概率即为 $\prod p_i + \prod (1 - p_i)$ ，我们要最小化这个式子，根据一些高中或带学数学的分析，可以得出当 $p_1 = p_2 = \dots = p_n$ 时，该式子取最小值，此时每句唱成功的概率计算可以得到，是 $\frac{2^n - 2}{2^n}$ 。
- 不会逆元的同学赶快去学呀，这种推个式子输出逆元的题很常见的！

## D. 牛牛做数论

- 本题答案的结论是：
- 问题一的答案为 $2$ 、 $2 \times 3$ 、 $2 \times 3 \times 5$ 、 $2 \times 3 \times 5 \times 7$ .....这些前若干个素数的积中，最大的且不超过 $n$ 的那一个，如 $n=233$ ，则答案为 $2 \times 3 \times 5 \times 7 = 210$ 。
- 问题二的答案为 $[2, n]$ 中最大的素数。
- 实现细节：对问题一，因为很少的一些素数前缀积就会超过 $10^9$ 了，因此预处理出前几十个素数即可，小心溢出问题；对问题二，从 $n$ 开始依次递减地暴力判断是否为素数即可，这是因为 $10^9$ 以内最大的两个素数间隔是282，所以这样做最多只要判断282个数字，且判断时往往跑不满根号的复杂度。

## D. 牛牛做数论

- 问题一：由于  $\phi(x) = x * \prod (1 - \frac{1}{p_i})$ ，之中  $p_i$  取遍  $x$  的所有种类的质因子（注意是种类而不是个数，如素因子 2 出现两次在公式里也只乘上一次），则  $H(x) = \prod (1 - \frac{1}{p_i})$ ，显然乘的项数越多、每一项越小  $H(x)$  就越小，而按照 2, 3, 5, 7, 11 ... 的顺序取  $p_i$  就可以达到这一目标。
- 问题二：对于素数  $p$  有  $\phi(p) = p - 1$ ，因此  $H(p) = \frac{p-1}{p}$ ，直觉可以感到这是挺大一数（毕竟  $H(x)$  最大不超过 1），所以可以猜出取最大的素数  $p$  即可，证明留作家庭作业。
- （相信出题人，他真的有证明过）

## B. 炸鸡块君与FIFA22

- 首先你需要掌握倍增 (ST表) ;
- 注意到起始分数若在 $\%3$ 意义下相等, 则经历 $[l, r]$ 一段后分数的**变化量**是一个定值;
- 定义 $st[3][200010][21]$ 之中 $st[k][i][j]$ 表示在初始分数为 $k$ 的情况下经历了 $[i, i + 2^j - 1]$ 一段儿游戏后分数的变化量;
- 假设预处理出了 $st$ , 则对于一次询问, 可以先将其初始分数 $s$ 对3取模, 然后按照倍增的套路从 $l$ 跳若干个2的次幂跳到 $r$ , 跳的时候要按照分数对3取模的结果来决定访问哪个 $st$ 值。



## B. 炸鸡块君与FIFA22

- $st[k][i][j]$ 如何预处理:
- 首先, 按照字符串内容初始化 $j = 0$ 的情况;
- 然后,  $st[k][i][j]$ 由 $st[k][i][j - 1]$ 和 $st[k][p][j - 1]$ 计算得到, 之中 $p = 2^{j-1}$ 。
- 转移方程:
- $st[k][i][j] = st[k][i][j - 1] + st[(k + st[k][i][j - 1]) \% 3][p][j - 1]$
- 之中 $(k + st[k][i][j - 1]) \% 3$ 表示若在 $i$ 处初始分数为 $k$ , 那么到了 $p$ 处时分数对3取模得多少

## B. 炸鸡块君与FIFA22

- 开始后20min的B题情况:

B	炸鸡块君与FIFA22	0/205	未通过
---	-------------	-------	-----

- 看了一下基本都是T的，不会算复杂度的小白们快去学习算法复杂度呀，或者起码花几分钟了解一下，性价比很高的；

## K. 冒险公社

- 一个其实比较无聊的+写起来比较麻烦的+看上去会被误认为可以贪心的DP问题;
- $dp[i][j][k][l]$ 表示考虑到前 $i$ 个字符, 当前结尾的三个字符  $(i, i-1, i-2)$  分别放了 $j, k, l (j, k, l \in \{0, 1, 2\})$ 时最大的绿岛数;
- 转移: 在求 $dp[i][j][k][l]$ 的时候考虑所有形如 $dp[i-1][\cdot][j][k]$ 的状态, 判断 $(j, k, l)$ 是否满足 $s[i]$ 的限制;
- 如果讨厌整很多维度的话, 也可以采用 $dp[i][j]$ 的形式来定义dp数组, 之中 $j \in [0, 26]$ , 表示原来的 $(j, k, l)$ 按三进制在一起组成的新整数,

# G. ACM is all you need

- 反过来考虑，对于位置  $i$ ，可以发现能够使得变换后  $f_i < \min(f_{i+1}, f_{i-1})$  的  $b$  值一定是连续的一段（但可能会到正无穷，这种情况可以规定一个比较大的数代替正无穷），如  $[6, 10, 6]$  中 10 的位置所对应  $b$  值的取值区间是  $[9, +\infty]$ ；
- 于是，我们对于每个  $i$ ，可以求出区间  $[l_i, r_i]$  表示若  $b$  在这个区间里取值就可以使得位置  $i$  满足条件；
- 我们发现，这样问题其实就转变成了给出  $n - 2$  个区间，求被区间覆盖最多的点被覆盖的次数（令  $b$  取该点的值，则覆盖该点的区间所对应的位置都可以取到最小值），这是一个比较经典的问题，可以通过对区间端点排序后遍历解决。

# 致谢

- 感谢qcjj和验题人（右）的帮助；
- 感谢做题的小白们来捧场~~~

名次	参赛者	学校	通过	罚时	A 9 15	B 9 12	C 9 38	D 10 14	E 12 24	F 8 13	G 5 12	H 9 21	I 8 10	J 9 17	K 7 17	L 12 13
1	zankizero	安徽省马鞍山市第二中学	12 AK	7817	869	915	932 (-5)	661	251 (-1)	681	1006 (-5)	280 (-2)	264 (-1)	258 (-1)	1096 (-3)	239
2	谢天意	阜阳师范大学	12 AK	195190	13395	13366	13482 (-4)	13335 (-2)	13292 (-1)	29182 (-2)	29105	13315 (-1)	13486 (-1)	13281 (-3)	16404	13261
3	积极的防守者	天津大学	12 AK	197634	16378	16389	16408 (-3)	16417	16422	16432 (-1)	16494 (-1)	16500 (-1)	16507	16513	16522	16526
4	fallleaves01	北京理工大学	12 AK	229392	19032	19041	19047	19071	19074	19092 (-1)	19140 (-1)	19152	19160	19169	19183	19186
5	FriedChicken	北京理工大学	12 AK	383292	33464	31970	33464	33465	33466 (-1)	33467	16634	33467	33468	33468	33466	33468
6	smzzi	北京理工大学	11	266531	23276 (-1)	23296	23308 (-1)	23326	23330	23347		23829	23836	23845	27524 (-2)	27528
7	快要秃头的Tíme	浙江科技学院	10	298041	24612 (-5)	28795	29083 (-11)	30369	30361 (-1)			30461 (-3)	30473	30401 (-1)	32559 (-5)	30381 (-1)
8	tokitsukaze	浙大宁波理工学院	8	268295	33326		33569 (-3)	33596	33443 (-1)	33493		33538 (-4)		33560 (-2)		33566
9	20203246	黑龙江大学	6	47684	19166			705 (-1)	716			7455 (-1)	19217	(-1)		382
10	TongWentao	浙江科技学院	5	150586		30911	30805 (-2)		30402 (-3)					29206		29160
11	ZincSabian	杭州电子科技大学	5	158066		31615 (-3)		31524 (-1)	31495	31661 (-1)						31670
12	哦是王毅呀	哈尔滨学院	2	64043					33523 (-4)							30439