

Contest 1 by Abracadabra

预期难度

- $B < F < CD < AE$

- 通过情况

- | | A | B | C | D | E | F |
|---|----|-----|-----|-----|---|----|
| • | 2 | 15 | 14 | 13 | 0 | 15 |
| • | 27 | 136 | 114 | 124 | 7 | 48 |

B - The Limit

- 这题只看大家还记不记得**洛必达法则**
-不该出成数学题的，我错了

关于洛必达法则

- Forum中有提到 $x / \theta (x \rightarrow \theta)$ 应该是 θ 还是INF，事实上 θ 可以认为是最高阶的无穷小， x / θ 应当输出INF（另一方面， θ/x 当然输出 θ ）
- 对这个式子洛必达法则后会变成 $1/\theta$ ，仍是INF
- 题目保证了不会分子分母同时为常数 θ ，所以不必担心洛必达进入死循环之类的.....

B. The Limit

- 每个用例包含两行字符串和一个整数，每个字符串最大长度为 5×8 (x的2次到9次幂)+3(x的1次幂)+2(常数)+1=46。
- 处理字符串s: 系数、x、指数、符号都是一字符，一个一个读就好。int a[10](置0)存储不同幂次对应系数，int flag存储符号系数，int coe(初始化1)存储系数绝对值
 - s[i]=='-' , flag = -1, coe = 1; s[i]=='+' , flag = 1, coe = 1. 初始化flag = 1(第一项为正时不带正号)
 - '0' <= s[i] <= '9' , coe = s[i] - '0'. 判断s[i+1]:
 - if(s[i+1] != 'x') a[0] = coe;
 - s[i]=='x'. 判断s[i+1]:
 - if(s[i+1] != '^') a[1] = coe;
 - else{ a[s[i+2] - '0'] = coe; i += 2; }

B. The Limit

- 系数为1时不会显示，故应当初始化coefficient为1，每当遇到 '+'、 '-' 时都应将coefficient置1。非常数项时应在读到 'x' 时再处理系数。
- 指数为1时不会显示，只需判断 'x' 后是否为 '^'
- 常数项只需判断数字后是否为 'x'

B. The Limit

- 求出每个字符串对应的数，最大情况下为 $3.92263245e9$ ，应用11存储。
- 分母不为0时，辗转相除，求最大公约，约分后输出，注意负号应放在分子前。
- 分母为0时，若分子不为0，输出INF；分子为0时，要用洛必达法则处理数组.....

F. Company Party

- 公司要举办派对，邀请员工参加
- 邀请的员工不能有直接的上下级关系
- 问最多可以同时邀请多少员工，以及不同的方案数

F. Company Party

- 用一个map把员工的名字映射到索引号
- 在索引号上建图
- 注意图中的树可能不止一颗，在所有出现的员工中，必定有至少一个没有上级
- 需要添加一个虚拟的big boss，向没有上级的员工连边

F. Company Party

- 树形动规
- 编写dfs函数，以节点为参数，每次调用函数得到以当前节点为根的子树的答案
- 得到答案之前，对其子节点进行递归调用，综合子节点的结果进行计算
- 在big boss上调用，得到最终答案

F. Company Party

- $\text{cnt}[u][0] = \sum \max(\text{cnt}[v][0], \text{cnt}[v][1])$
- $\text{cnt}[u][1] = 1 + \sum \text{cnt}[v][0]$

- $\text{way}[u][1] = \prod \text{way}[v][0]$
- $\text{way}[u][0]$ 初始为1, 对每个子节点v分情况
- 若 $\text{cnt}[v][0] > \text{cnt}[v][1]$, 乘 $\text{way}[v][0]$
- 若 $\text{cnt}[v][0] < \text{cnt}[v][1]$, 乘 $\text{way}[v][1]$
- 若 $\text{cnt}[v][0] = \text{cnt}[v][1]$, 乘 $\text{way}[v][0] + \text{way}[v][1]$

C. Shot Boy

- 本题第一眼感觉可以用暴力深搜来做，但是因为 $N \leq 100$ ，那总共就有一万个点，如果这样子深搜，肯定会超时。
- 那我们该怎么解决这题呢？

C. Shot Boy

- 我们注意到，如果一个空地上有男孩，那么，这空地上的这一行，直到墙为止，都不能有其他男孩，同样，这一列也是。
- 那么，我们可以把这样的一行记做一个点 A_i ，同样，这样的一列也记做一个点 B_j 。如果这一行这一列有交点，那么我们认为 A_i 和 B_j 之间有一条连线。
- 我们把所有的 A_i 都放在一个集合A中，把 B_j 都放在一个集合B中。A中任意两点不可能有连线，B中也是。而 A_i ， B_j 中可能有连线。
- 此时，这道题就变成一道二分匹配的题了。

C. Shot Boy

- 我们只要求出最多可以从A, B中取出多少点, 使得这些点中, 两两间没有连线。
- 可以用网络流或者匈牙利算法, 求出AB间最多有多少条线, 每两条线两端的端点, 都互不相同。然后只要把A, B中所有点的个数, 减去最多连线的条数, 就可以得到答案。

D. Bulbasaur and Matrix

- 对于“使最小值最大”的题目，容易想到用二分答案。
- 如何检验呢？
- 要构成一个四角坐标为 $(x1, y1)$ $(x1, y2)$ $(x2, y1)$ $(x2, y2)$ 的矩形，需要在不同的行里，出现至少两次 $(y1, y2)$ 这样一对列坐标。
- 那么列坐标最多有多少对呢？ m^2 对。
- 所以，对于每一行，我们暴力枚举列坐标对。一旦发现某个列坐标对出现了两次，就说明构成了矩形，检验成功。
- 复杂度 $O(m^2 \log(MAXA))$
- Source:
<http://codeforces.com/problemset/problem/333/D>

E. Balls on Rail

- 与经典的链上蚂蚁碰撞问题相似。
- 将球体碰撞视为交错而过，仅交换了各自的标号。我们可以求出所有 n 个最终位置，并将某一个球和它的位置对应起来，之后就能根据所有球相对顺序不变的性质确定所有球的位置。
- 假设我们试图将最初具有标号 i 的球与其最终位置对应。我们需要考虑的是这个球在运动过程中交换了多少次标号。球 i 只会与方向相反的球相撞。我们首先求出 i 绕轨道运行了多少圈，不足一圈的部分需要依次遍历统计。
- Source:
<http://codeforces.com/contest/652/problem/F>

A. MX's RPG Game

- 题目大意：给一个 n 点 m 边带权无向图，从起点到终点任选一条路径，最多可以选择 k 条边不支付其权值，求路径上(最大代价-最小代价)的最小值。

A. MX's RPG Game

- 依次枚举需要支付的最大边最小边，假设这就是最终的答案区间，把其中的边权值设为 0，其余设为 1，看终点和起点的距离是否小于等于 k ，如果小于等于 k 就可以更新答案了。外面枚举了 m^2 层。
- 改进：从大到小排序边后，如果已经枚举了第 a 条到第 b 条的区间，并且得到此时起点和终点距离小于等于 k ，枚举第 $a + 1$ 条边为最大边时，最小边直接从第 b 条边开始枚举，而不需要从第 $a+1$ 条后面那一条开始。这样外面的枚举层数是 $O(m)$ 。

A. MX's RPG Game

- 这样做是正确的。枚举第 $a+1$ 条为最大边时，由于将第 a 条边的权值由 0 变成了 1 ，始末点的距离不减，所以至少需要从这条边开始枚举。
- 关于如何得到始末点距离：直接spfa是可以过的。也可以将权值为 0 的边两端点缩起来。然后bfs。 $O(m)$
- 总时间复杂度 m^2