Summer 2017 - Contest 7 Solution

Lucky String

由于判断一个字符串是否幸运只和它的子串是否幸运有关，所以这个题里所有幸运字符串都是等价的。

只要重复地把 1001 都变成 0，直到最后不能再变为止，看一下最后的串是不是 0 就知道啦~

用栈来维护即可 $O(n)$ 完成。

Equation

从最小的那一位（以下称为第一位）开始，如果 a 和 b 的第一位都是 0，那么 x 的第一位只能是 0（可以试一下如果 x 的第一位是 1 那么不能满足式子）。

继续分析我们发现如果 a 和 b 的第一位都是 1，那么 x 的第一位还是 0，但是两个 1 加起来向前进了一位；如果 a 和 b 的第一位不一样，那么 x 的第一位是1，不进位。

然后我们分析第二位，如果第一位没有进位那么分析过程和上面一样。如果有进位，则 (0, 0) --> 1 且进位，(1, 0) --> 0 且进位，(0, 1) --> 0 且进位，(1, 1) --> 1 不进位。

这样一位一位地分析，就能得出 x 了。我们可以发现满足式子的 x 最多只有一个。复杂度 $O(TlogA)$

那什么时候无解呢？举个例子，考虑 $a = 1$ 和 $b = 1$：

第一位 (1, 1) --> 0 且进位
第二位 (0, 0) --> 1 且进位
第三位 (0, 0) --> 1 且进位
…

就是这种到后面会不停地进位的数，就无解了。

Probability

设 i 号同学在某一局赢得游戏的概率 $w_i = p_{1,1} \times p_{1,2} \times \cdots \times p_{1,m}$，某一局没有人获胜的概率为 $w_0 = 1 - w_1 - w_2 - \cdots - w_n$。

那么 1 号同学获胜的概率为 $w_1 + (1 - w_0)w_1 + (1 - w_0)^2 w_1 + \dots$

这样我们就需要对一个公比 $q < 1$ 的数列进行无穷多项的求和，我们都知道公式应该是 $\frac{a_1}{1-q}$，所以上面的式子可以化简为 $\frac{w_1}{1-(1-w_0)} = \frac{w_1}{w_0}$

把式子展开后我们发现，这个问题等价为以下问题：

已知 n 和 $w_1, w_2, \dots, w_n$ 的值，求 $\frac{w_1}{w_1+w_2+\cdots+w_n}$，其中 $w_i = p_{i,1} \times p_{i,2} \times \cdots \times p_{i,m}$.

于是我们就开心地把所有的数字加加乘乘，一交发现 WA 了，这是为什么呢？

因为 double 的精度仅到小数点后 16 位，如果数据是 10000 个 0.1000 相乘，这个精度是 double 存不下的。

不过由于答案只要求输出小数点后 6 位，所以我们只需要把每个数最前面的几个非零数字记录下来即可（后面的数字对运算结果的影响很小，不会在小数点后 6 位体现）。但是每个数第一个非零数字的位数是不一样的，怎么办呢？我们再记录每个数字的精度是 10 的负几次方即可。

这样，我们把所有的数都变成 $a \times 10^{-b}$ 的形式，记录下 b 最小是多少，之后进行运算就能避过精度的问题了。

还可以使用 Python 过题美滋滋

```
from __future__ import division

tcase = int(raw_input())

for _ in xrange(tcase):
    n, m = map(int, raw_input().split())
    a, b = 0, 0
    for i in xrange(n):
        t = reduce(lambda x, y: x * y, map(lambda x: int(float(x) * 10000 + 1e-7), raw_input().split()))
        if i == 0:
            a = t
        b += t
    print('%.9f' % (a/b))
```

Tetris Puzzle

X	X	O	X	X	X	O	X	X	X	O
X	O	X	X	X	O	X	X	X	O	X
O	X	X	X	O	X	X	X	O	X	X
X	X	X	O	X	X	X	O	X	X	X
X	X	O	X	X	X	O	X	X	X	O
X	O	X	X	X	O	X	X	X	O	X
O	X	X	X	O	X	X	X	O	X	X
X	X	X	O	X	X	X	O	X	X	X
X	X	O	X	X	X	O	X	X	X	O
X	O	X	X	X	O	X	X	X	O	X
O	X	X	X	O	X	X	X	O	X	X
X	X	X	O	X	X	X	O	X	X	X

Possible Roots

给定一棵边权树及K个特殊点s_i，求出以多少点为根满足所有特殊点高度等于给定值h_i。

从减少判定代价的方向入手，根据一对特殊点uv之间的距离及其h_i即可求得可行解的一个范围，对这个范围内任意点w有$dist(w,u) = dist(w,v) + (h(u) - h(v))$，只需测试其中一个点，将$(s_0, s_i)$这$n-1$个结果求交后即将判定代价减少到$O(1)$。求交部分可以用DFS序简化，范围求取需要树上倍增或其他方法

Pair of Path

给定一个 DAG ($n \le 5e3, m \le 2e4$)，求总长最小的两条边不相交路径 A-B C-D。

用 $dp[i][j]$ 表示路径 A-B 到达 i，路径 C-D 到达 j 的最小代价，其中 i, j 为点的拓扑序标号。为了保证路径不相交，每次转移需要保证更新后的那一维在某个序上比另一维更大，很自然地想到用拓扑序。因为需要处理 $dp[i][i] - dp[i][j] - dp[j][j]$ 的非法转移，需要将状态修改为 $dp[i][j][type]$ 来表示边 $i - j$ 是否被使用。由于边集共出现 n 次，复杂度为 $O(nm)$。

Circular Shift

后缀自动机

Stars in a Can

给定三维空间中一些点，问最小包围圆柱的体积.要求一侧底面至少有三个点 (n <= 1000)
求三维凸包，枚举每个面作为底面，投影完之后求最小包围圆即可.O(n^2logn)