

2012 年华南理工大学“TCL 通讯杯”程序设计竞赛简单题解报告

A 题

某年的 11 月 17 日是星期几。

模拟。注意闰年等 **trick**

PS: 模板里好像有个日期类

B 题

这题本来是最裸的 **LCA**（最近公共祖先）。

但是只有一个询问，所以直接 **while()**...

模拟。对两个点不断往上找直到第一个父亲为止。

C 题

给你提交的结果问最后的排名；

模拟。唯一的 **trick** 可能是 **AC** 之后还有提交，这种提交应该无视。

D 题

字典树 + 乱搞，注意各种 **trick**，例如有多个符合的选第一个，**Invalid** 情况等

E 题

$$\text{Hash}(n) = \text{ascii}('()') \prod_{i=1}^n (a^{2^i-1} + 1) + (\text{ascii}('()') - \text{ascii}('()')) \prod_{i=0}^n (a^{2^i} + 1)$$

连乘的项是有循环的。**MOD 1000000000** 的循环节太长，可以拆成 **MOD 2^8** 和 **MOD 5^8** 分别求解。

F 题

把一次变换看成一个矩阵，用快速幂求出 **k** 次以后的变换矩阵。

然后类似解方程组，得到一个解空间

在这个解空间上用 **polya** 定理，得出本质不同的解。

G 题

最小费用流。

因为包含负环所以不能直接用 **spfa** 跑最小费用流。

所以消去负圈后 **spfa** 跑最小费用流。

H 题

以 **price** 作为指数，**score** 作为系数构造多项式型母函数。问题转化成求一个多项式的 **14** 次幂，然后再求结果系数的前缀和。求多项式的 **14** 次幂时需要使用 **FFT**。

I 题

题目很长，本质是一道很水的模拟题，只要耐心的把题目读完，依据题目的描述来处理各种买卖和取消的情况就 OK 了，基本没什么坑爹的 trick。

J 题

暴搜+剪枝。

枚举最小长度的值，令 maxlen 为给定的棒子堆中最长的棒子， sumlen 为这堆棒子的长度之和，那么取值必定在范围 $[\text{maxlen}, \text{sumlen}]$ 中，然后就是剪枝。

设枚举值为 val ，降序处理

剪枝：

- 1、由于所有原始棒子等长，那么必有 $\text{sumlen} \% \text{val} == 0$;
- 2、若能在 $[\text{maxlen}, \text{sumlen} - \text{val}]$ 找到最短的 val ，该 val 必也是 $[\text{maxlen}, \text{sumlen}]$ 的最短；若不能在 $[\text{maxlen}, \text{sumlen} - \text{val}]$ 找到最短的 val ，则必有 $\text{val} = \text{sumlen}$;
- 3、由于所有棒子已降序排序，在 DFS 时，若某根棒子不合适，则跳过其后面所有与它等长的棒子；
- 4、对于某个目标 val ，在每次构建新的长度为 val 的原始棒时，检查新棒的第一根 $\text{stick}[i]$ ，若在搜索完所有 $\text{stick}[]$ 后都无法组合，则说明 $\text{stick}[i]$ 无法在当前组合方式下组合，不用往下搜索(往下搜索会令 $\text{stick}[i]$ 被舍弃)，直接返回上一层。

K 题

记忆化搜索。

设 $\text{dp}[x][y]$ 为到该位置的最大距离，设所给矩阵为 $\text{d}[x][y]$ 。

则 $\text{if}(\text{d}[x-1][y] > \text{d}[x][y])$

$\text{dp}[x][y] = \max(\text{dp}[x][y], \text{dp}[x-1][y] + 1);$

$\text{if}(\text{d}[x+1][y] > \text{d}[x][y])$

$\text{dp}[x][y] = \max(\text{dp}[x][y], \text{dp}[x+1][y] + 1);$

$\text{if}(\text{d}[x][y+1] > \text{d}[x][y])$

$\text{dp}[x][y] = \max(\text{dp}[x][y], \text{dp}[x][y+1] + 1);$

$\text{if}(\text{d}[x][y-1] > \text{d}[x][y])$

$\text{dp}[x][y] = \max(\text{dp}[x][y], \text{dp}[x][y-1] + 1);$

注意下边界问题就可以了。