

# 寒假集训模拟赛(二)

(题解)

## 1. 投票

(vote.cpp/c/pas)

来源：NOIP2014 普及组 T2 比例简化

第一眼看到是不是以为要用数论知识啊？

不会，也没关系~~~ K 的数据范围已经暴露了一切，暴力枚举  $A'$ ， $B'$  即可。

当然，判断互质需要用到欧几里得算法，在 K 的范围内枚举判断就 ok！

不会欧几里得算法怎么办？这个时候脑补算法就派上用场了。设有一对不互质的  $A_1$ ， $B_1$  被枚举到，那么，在  $A' = 1..K$ ， $B' = 1..K$  的枚举顺序下，一定存在一对  $A_2$ ， $B_2$ ，使得  $A_2$  与  $B_2$  互质， $A_2/B_2 = A_1/B_1$ （这样可以推得  $A_2 = A_1/\gcd(A_1, B_1) < A_1$ ， $B_2 = B_1/\gcd(A_1, B_1) < B_1$ ），所以  $A_2$  与  $B_2$  一定已经先于  $A_1$ ， $B_1$  被枚举到。这样，只要把更新条件设为  $>$ （而不是  $\geq$ ），就可以在不判断互质的情况下得到正确的解。

## 2. 活动

(activity.cpp/c/pas)

**0 分（错误）算法：**要求子串长度至少为 3，所以每三个点分割一次，状态转移方程为：

$$f(x, y) = \begin{cases} f(x-1, y-1), & \text{不能加入公共子序列} \\ \max(f(x-1, y-1), f(x-3, y-3) + 3), & \text{可以加入公共子序列} \end{cases}$$

错误之处在于每三个字符分割一次则可能的最长公共子序列中的长度不是 3 的整数倍的子串无法被正确计入。

**20 分算法：**这道题比较难想状态转移方程，怕答案错误的话，就上暴力吧，二进制子集枚举，时间复杂度  $O(2^n)$ 。

**80 分算法：**枚举分割长度  $k(k \geq 3)$ ，状态转移方程为：

$$f(x, y) = \begin{cases} f(x-1, y-1), & \text{不能加入公共子序列} \\ \max(f(x-1, y-1), f(x-k, y-k) + k), & 3 \leq k \leq x, y, \text{可以加入公共子序列} \end{cases}$$

注意一个细节，在枚举  $k$  的时候，如果两个字符串中新加入的最后一个字符不相同，那么如果  $k$  继续增大，

则生成的子串也不可能再加入公共子序列，所以要加入一句“else break;”，对于随机数据，这种操作的表现非常好，足以得到 80 分，而不加则只能得到 30 分，时间复杂度  $O(n^3)$ 。

**100 分算法**：对于非随机的数据，因为可能存在  $2000 \times 2000 \times 2000$  的极端情况，所以应该预处理以求继续降低复杂度。时间复杂度  $O(n^2)$ 。

### 3 . 逃学计划

(plan.cpp/c/pas)

“不自由，毋宁死。”——帕特里克·亨利

**40 分算法**：直接 DFS 即可。

**70 分算法**：图的点数少，边数多，所以路径上可能有很多重边。加入最优性剪枝策略，即如果当前状态下的  $a, b, c$  之和已经大于等于当前搜到过的最优解，继续往后搜索也不会得到更好的结果，剪枝，即可在较短时间内出解。

**100 分算法**：虽然点数多，但受数据范围中通往终点的点入度很小的启示，不难想出可行性剪枝策略，即先从终点倒着 BFS，把所有不能到达终点的点屏蔽。或者存反图 DFS 也能解决问题。由于数据量大建议使用读入优化。

对于 BFS，虽然是可以得到正确结果的，但是由于  $n \leq 10000$ ，可以比较安全地使用 DFS，代码复杂度下降，而且时间效率也相对较高。