# 2013 Multi-University Training Contest 7 Solution

Fudan University

2013 年 8 月 13 日

## Problem A. Hyperspace

考虑点 A 和点 B 的曼哈顿距离 $S = \sum_{i=1}^{k} |A.x_i - B.x_i|$。

有 $A.x_i - B.x_i, B.x_i - A.x_i \le |A.x_i - B.x_i|$，所以当符号错误时，不会使答案变小。考虑 $2^k$ 枚举绝对值的符号，显然有一种符号能取到最值。

我们用 $2^k$ 个堆维护每种符号下 $\pm A.x_i$ 的最小值和最大值，每次从 $2^k$ 个堆中取出最大值和最小值，相减，取最大差。这个差就是当前询问的答案。

## Problem B. Building Fence

没想到出成原题，导致部分队伍用网上的代码过了，实在抱歉。

基本思想就是求出所有的"关键点"。将两圆两两求外切线，得到所有切点；将圆和三角形的三个点依次做切线，也得到切点；加上所有三角形的三个点。

对这些点用 Graham 求凸包，对于凸包上的每条边，如果它们在同一个圆上，用等价的圆弧替代即可。

## Problem C. Finding string

先对每个字符串循环节，由于循环节长度不是很大，所以可以求出模式串在该循环节中出现次数。

然后考虑跨串的情况。对考虑每个循环串，记录前面的串的后 $|P| - 1$ 位，和该循环串的前 $|P| - 1$ 位，合并后 KMP 求出模式串出现次数累加到答案。

## Problem D. Mutiples on a circle

问题其实就是问在圆环上有多少种方案可以使得选出来的一段是 $K$ 的倍数。

那么如果我们知道了 $i$ 为结尾的所有段，我们可以将这些段按照 $\bmod K$ 的余数分类，记为 $count[i][r]$，$r$ 表示余数。那么如果我们在段末尾加上 $i+1$，我们就可以仅根据余数来确定这些段在添加了 $i+1$ 之后 $\bmod K$ 的余数是多少 $(r \times \exp(10, digits[i+1]) + number[i+1])$。然后只要减去刚才 $i+1$ 到 $i$ 那一段，再加上单独 $i+1$ 的一段就可以了。

如此便可以在 $O(nK)$ 的时间内统计出答案。

## Problem E.   Cube number on a tree

分治。

用三进制压缩状态表示每个点包含的每个因子个数除以 3 的余数。

每次找到树的一个重心，把树分成若干棵子树，统计跨过重心的合法链条数，再递归处理每一棵子树。

统计跨过重心的合法链条数：

计算出子树内每个点到对应子树的根的压缩状态，对于每个点的压缩状态，求其他几棵子树内有多少个点对应的到子树根的压缩状态与该点到重心的压缩状态合并得到的压缩状态为 0。可以每次用 map 维护。

## Problem F.   Backup Plan

题意：有 $N$ 台服务器和 $M$ 个数据库，每个数据库在所有服务器上都有一份拷贝，查询的时候按照一定顺序查找直到找到一台可用的服务器。求一个顺序，使得服务器完好和任意一台服务器坏掉的时候负载均衡。

正确理解题意之后可以发现，只有顺位是 1 的和顺位是 2 的服务器有意义，剩下的随便输出即可。对于不坏的情况，直接均匀分配即可。

考虑坏掉一台的情况，我们把这台服务器在不坏的情况下承载的所有数据库分到其他服务器上即可，优先放到承载数据库数量较少的服务器上。容易发现，由于每台服务器坏掉的情况是独立的，对每台服务器做这样的操作后就得到了一个合法的答案。

其他：$n = 100$ 的意图是让人暴力。

## Problem G.   Present Day, Present Time

题意：有 $N$ 堆石子和 $M$ 个石子回收站，每回合操作的人可以选择一堆石子，从中拿出一些放到石子回收站里（可以放进多个回收站，每个回收站可以使用无数次），但每个石子回收站每次只能接收特定数量的石子。不能操作的输。如果有人操作完之后，有任意一堆石子无法完全回收，那么他直接输。

一个显然的结论是，每个游戏的 SG 值就是用 $M$ 个回收站，完全回收这堆石子可行的最大操作次数。由于最大的 $B_i$ 比较小，立方暴力背包即可（比较显然的是，$\max B_i{}^2$ 以上的周期是 $\min B_i$）。

## Problem H.   Pirate's Chest

先预处理出每层拿到道具 1、拿到道具 2、都拿到的代价，拿一个道具可以直接 spfa，拿全全部道具的话是一个 3 个点的 steiner-tree 问题，可以枚举中间点求解。

然后不难发现打开所有箱子消耗的 HP 随上的层数增加单调不增（显然多一层只是多了一些选项，严格不亏）

然后二分上的层数 p，求出只用前 p 层打开所有箱子所需的最小代价。

构建最小割模型

两排点，左边一排代表钥匙，右边一排代表撬棍，设 $A_i$ 表示钥匙 i 对应的点，$B_i$ 表示撬棍 i 对应的点

对于每个钥匙 $i$，若 $i$ 所在层只有一个工具，则从 $S$ 到 $A_i$ 引一条边容量为获得 $i$ 的代价。

对于每个撬棍 $i$，若 $i$ 所在层只有一个工具，则从 $B_i$ 到 $t$ 引一条边容量为获得 $i$ 的代价。

若某一层有两把钥匙，新建一个点 $p$，从 $S$ 到 $p$ 连一条边容量为获得两把钥匙的代价，从 $p$ 到两把钥匙分别连一条边容量为单独获得这一把钥匙的代价。

若某一层有两把撬棍，新建一个点 $p$，从 $p$ 到 $T$ 连一条边容量为获得两把撬棍的代价，从两把撬棍到 $p$ 分别连一条边容量为单独获得这一把撬棍的代价。

对于每个宝箱 $i$，从 $A_{ai}$ 到 $B_{bi}$ 连一条边容量为 $D_i$ 的边。

不难发现该图的每个割都对应一个打开 $n$ 个箱子的方案，所以最小割即为最小代价。

## Problem I.  Trip Advisor

通过观察不难发现一条性质：输入图的每个 BCC 都是一个环，并且如果将每个环都看成一个点的话，整个图形成了一个树的结构。

因而先将给定图缩 BCC，对于一条询问 $(u, v, p)$:

1. 若 $u = v$，无解当且仅当 $p$ 不等于 $u$。
2. 令 $cu$, $cv$, $cp$ 表示缩点之后的树上 $u$, $v$, $p$ 所在环所对应的点，若 $cp$ 不在树上 $cu$ 到 $cv$ 的路径上，无解。
3. 若 $cu = cv$，则 $cp$ 必须等于 $cu$，否则无解。
4. 若 $cu = cp$（$cv = cp$ 同理），无解当且仅当从 $u$ 朝着 $v$ 走一步就不在环 $cu$ 中了，并且 $p$ 不等于 $u$。
5. 否则，无解当且仅当路径经过环 $cp$ 时，进出是一个点，并且那个点不是 $p$

执行以上算法可以判定可行性，相关的一些判断（如一个点是否在树上某两点间的路径上）用倍增 lca 解决。

## Problem J.  GCD of Sequence

比赛中很长的时间数据有误，主要是手贱有两组数据存在 $a_i > M$，导致一些队伍 WA 在这两组。抱歉。

考虑一个容斥. $f_d$ 表示满足上述条件的最大公约数为 $d$ 的方案数（就是答案）

$g_d$ 表示满足上述条件的公约数为 $d$ 的方案数.

显然有 $g_d = \sum\limits_{d|n} f_n$

考虑 $g_d$ 如何计算.. 假设当前数列 $a$ 有 $x$ 个是 $d$ 的倍数，那么:

1. 如果 $n - x \le K$, 我们可以把那些不是倍数的 $n - x$ 个数变为 $d$ 的倍数. $\lfloor \frac{M}{d} \rfloor^{n-x}$, 还剩下了 $K - (n - x)$ 个需要改动.. 所以还要乘上 $(\lfloor \frac{M}{d} \rfloor - 1)^{K-(n-x)}$
2. 如果 $n - x > K$ , 那么 $g_d = 0$

利用 $g_d$ 就可以求出 $f_d$..

求的过程是利用 $O(\frac{m}{1} + \frac{m}{2} + \frac{m}{m} + \ldots + \frac{m}{m}) = O(m \log m)$ 的方法.