2022 年上海市大学生程序设计竞赛参考题解

主办方：上海大学
题目作者：上海交通大学

2022 年 9 月 24 日

1 总览

题目难度估计

签到题 G, N

简单题 A, E, H

较简单题 L, M

中档题 B, C, *D*, I, J

较难题 *F*, *K*

排名分布

11 题 2 队

10 题 1 队

9 题 2 队

8 题 8 队

7 题 13 队

6 题 25 队

5 题 16 队

4 题 17 队

3 题 28 队

2 题 41 队

1 题 7 队

A. Another A+B Problem

作者 fstqwq　**一血** 19(+) 上海交通大学 梦想天生

题解　一共只有 $\binom{101}{2}$ 个合法等式，逐个判断即可。

B. Bracket Query

作者 fstqwq, suniyu, Tommyr7　**一血** 109(+) 上海交通大学 梦想天生

题解　我们把左括号看作 +1，右括号看成 −1，令 a_i 表示扩号序列在这种表示下的值，s_i 表示 $\sum_{j=1}^{i} a_j$，同时令 $s_0 = 0$。一个括号序列合法，当且仅当：

- $\forall i \in [n], |s_i - s_{i-1}| = 1$，且
- $\forall i \in [n], s_i \geq 0$，且
- $s_n = 0$。

给定的 q 个限制中，第 i 个限制可以表示为 $s_r - s_{l-1} = c_i$。将 $|s_i - s_{i-1}| = 1$ 放松为 $-1 \leq s_i - s_{i-1} \leq 1$，那么对以上条件运行差分约束是有解的必要条件。我们现在可以证明：反之亦然，即如果差分约束有解，则存在合法的括号序列。可以考虑以下方法证明：

1. 使用最短路在差分约束有解也即没有负环时，所有能参与最短路更新的边权权值均满足连接两端点标号差的奇偶性与边权奇偶性相同，因此不会出现 $s_i = s_{i-1}$ 的情况，因此 $s_i - s_{i-1} = a_i$ 的绝对值恰为 1，因此满足了括号序列合法的充分条件。
2. 我们可以考虑改变 s_i 的定义来更好地说明这个奇偶性。令 s_i 此时为前 i 个字符中左括号的数量。此时，$s_i - s_{i-1} = a_i \in \{0, 1\}$，而由于整数边权，最短路一定返回整数解，因此一定能满足充分条件。可以发现，这个差分约束和之前的差分约束是等价的。

如果直接运行差分约束，使用 Bellman-Ford 的复杂度高达 $O(nq)$。注意到，q 个限制中其实只有联通新的连通块的条件是有效的，因此除了至多 n 个条件有效，其余均是冗余的。使用带权并查集维护连通性并判断是否存在矛盾的条件，随后使用 Bellman-Ford 对 $O(n)$ 条边的图运行最短路判断出负环或者得出解，时间复杂度 $O(n^2)$。

我们注意到有队伍使用 SLF SPFA 对于 $O(nq)$ 的建图直接通过了。直观来讲，本题考虑在合法的情况下，如果所有 $(i, i+1)$ 连上的话，那么 SLF SPFA 是一个 0-1 BFS （此时复杂度为 $O(n+q)$）。由于构图特殊，并且要确保没有正解被卡常数，因此很难卡掉。

C. Coffee Overdose

作者 fstqwq, desprado2　**一血** 136(+1) 上海交通大学 吸风饮露

题解　**结论 1：**喝咖啡的时候体力值 u 必然满足 $u - C \le \frac{C(C-1)}{2}$。证明显然。

结论 2：如果从某个时刻开始喝咖啡，则此后喝咖啡是必然不间断的。证明如下：如果在第一次喝咖啡后存在某个时刻体力值为 i，可以喝咖啡却没有喝，则必然可以将体力值为 i 之前的第一次喝咖啡时刻推迟到 i 而其他喝咖啡时刻不变，这样调整后总完成度一定会变好。

倒过来考虑，如果入睡前最后一次喝咖啡时的体力为 x，则可以从 $\{x, x+C+1, x+2(C+1), ...\}$ 中找到最大的 k，满足 $x+k(C+1) \le min(\frac{C(C-1)}{2}+C, S)$，则从体力值为 $x+k(C+1)$ 开始喝咖啡，是最优的。k 是关于 x 的分段函数，最多可以分为两段。进而，可以发现总完成度也是关于 x 的分段函数，最多同样分为两段，且每一段都是关于 x 的二次函数。列出式子计算出最大值点即可。总时间复杂度 $O(1)$。

此外，本题在分段正确的情况下，每一段也可以使用三分法求最大值，总时间复杂度为 log 级别。另外还存在一些复杂度为根号级别的做法，此处不做叙述。

D. Demonstrational Sequences

作者 desprado2　**一血** N/A

题解　如果你熟悉 Pollard-rho 算法，那么你会发现这题的做法就是对 Pollard-rho 算法稍作修改。

对于题意中的无穷数列，有一个关键的性质是，对于一个固定的间隔 k ，定义 $d_i = x_{i+k} - x_i$，则对于任意非负整数 $u,\ v$ $(0 \le v < u)$ 必然有 $d_v | d_u$。证明如下:

$$\begin{aligned} d_{i+1} &= x_{i+k+1} - x_{i+1} \\ &= x_{i+k}^2 - b - (x_i^2 - b) \\ &= (x_{i+k} + x_i) \cdot (x_{i+k} - x_i) \\ &= d_i \cdot (x_{i+k} + x_i) \end{aligned}$$

因此对于数列 $\{d_i\}$，后一项总是前一项的倍数。

设数列 $\{p_i\}$ 表示无穷数列 x_i 每一项对 P 取模后的数列，数列 $\{q_i\}$ 表示无穷数列 x_i 每一项对 Q 取模后的数列。则原数列中某两项之差对 P 和 Q 求 GCD 的结果，可以转化为数列 $\{p_i\}$ 和数列 $\{q_i\}$ 中两项之差对 P 和 Q 求 GCD 的结果。证明如下:

$$\begin{aligned} \gcd(x_u - x_v, P) &= \gcd(P, (x_u - x_v) \bmod P) \\ &= \gcd(P, |x_u \bmod P - x_v \bmod P|) \\ &= \gcd(P, |p_u - p_v|) \end{aligned}$$

对 $\{q_i\}$ 同理。

现在考虑怎样判断一个无穷数列是有意义的。对于数列 $\{q_i\}$ 将 q_i 向 q_{i+1} 连边，则可以发现在经过若干步后会形成一个环。记出现环的第一个位置下标为 r ，环的大小为 len，则我们只需要判断 $\gcd(P, |p_{r+len} - p_r|)$ 是否为 Q 。若是，则已存在 $u = r + len,\ v = r$ 满足条件，原数列是有意义的；否则原数列就不是有意义的，证明如下:

$Q | x_u - x_v$ 是 $\gcd(x_u - x_v, P) = Q$ 的一个必要条件。下面证明，如果 $\gcd(P, |p_{r+len} - p_r|)$ 不等于 Q，则当 $Q | x_u - x_v$ 时必有 $\gcd(x_u - x_v, P) \neq Q$ 。首先，根据上面推出的性质，不难证明:

- 对于一个固定的间隔 k ,对于任意非负整数 $u,\ v$ $(0 \le v < u)$ 必然有 $\gcd(P, x_{v+k} - x_v) | \gcd(P, x_{u+k} - x_u)$
- 对于两个间隔 k_1 和 k_2 满足 $k_1 | k_2$，则对于任意非负整数 v 必然有 $\gcd(P, x_{v+k_1} - x_v) | \gcd(P, x_{v+k_2} - x_v)$

注意到 $\gcd(P, |p_{r+len} - p_r|) > Q$ 。显然，$Q | x_u - x_v$ 等价于 $v \ge r,\ len | u - v$ ，则根据上述两点性质，必有 $\gcd(P, |p_{r+len} - p_r|) | \gcd(x_u - x_v, P)$ 。于是 $\gcd(x_u - x_v, P) \ge \gcd(P, |p_{r+len} - p_r|) > Q$。因此原数列 $\{x_i\}$ 不存在 $u,\ v$ 满足条件，原数列不是有意义的。

综上所述，由于 q_i 最多只有 Q 种不同取值，所以 r 和 len 最坏情况下均为 $O(Q)$ 级别的，总时间复杂度是 $O(kQ)$。

E. Expenditure Reduction

作者 fstqwq　**一血** 12(+) 上海大学 江海寄余生

题解　维护一个 $|S| \times |\alpha|$ 的二维数组表示每个字符的下一个位置在哪，然后从 S 的每个位置出发尝试得到 F。时间复杂度 $O(|S| \cdot |\alpha| + |S| \cdot |F|)$。

或者可以考虑动态规划，$f_{i,j}$ 表示考虑到 S 的第 i 个位置，当前 F 匹配到第 j 位，从前向后维护即可，$\min f_{i,|F|}$ 即是答案。时间复杂度 $O(|S| \cdot |F|)$。

F. Forest of Magic

作者 AntiLeaf　**一血** N/A

题解　先不考虑加叶子的操作，也不考虑 c 的限制。

假设在路径 (x, y) 上进行了一次修改，设它们的 LCA 是 z，询问 u 时贡献只有 u 在路径上或者 u 是 z 的祖先两种情况，否则没有贡献。设每个点的深度是 d_i，容易发现贡献一定可以表达为 $a_i \times d_i + b_i$ 的形式，差分之后就可以表示为若干个单点修改，询问时只需子树求和即可。求一下 DFS 序，用一个单点修改区间求和的线段树维护就行了。如果加上 c 的限制，考虑到需要强制在线，可以用一个线段树套平衡树（省空间）维护。

加叶子可以转化为在 DFS 序列上插入一个点，同时仍然要维护区间求和。我们可以把外层也换成一个平衡树，也就是用平衡树套平衡树维护。注意外层的平衡树必须用重量平衡树，比如替罪羊树和 Treap。另外由于里面套的平衡树大小并不是一样的，因此判断失衡时要考虑上里面的平衡树的大小；以及替罪羊树重构时每次不能直接找中点，而是找到一个尽量均分两边套的平衡树大小之和的点作为根。

考虑到 n, m 同阶，复杂度可以写成 $O(n \log^2 n)$，空间 $O(n \log n)$。

考虑到平衡树套平衡树实在太难写（标程写了 12k），用块状链表代替或者直接定期重建也是可以的，复杂度大概是 $O(n^{1.5} \log n)$。

G. Gua!

作者 fstqwq　**一血** 9(+) 上海交通大学 梦想天生

题解　我相信大家读完题已经会了！但是重生不会，很可惜。

```
print("gua!" if D > (R * S // 60 + (1 if R > 0 else 0)) * B else "ok")
```

注意上下取整可能会有小细节。

H. Heirloom Painting

作者 desprado2　**一血** 18(+) 复旦大学 有向无环图状数组

题解　考虑最后一次涂色，必然会形成一个长度不小于 k 的相同颜色的连续区间。因此如果不存在长度不小于 k 的相同颜色的连续区间，则必然无解。不难发现，每一段相同颜色的连续区间都是独立的，假设其长度为 len，则至少需要 $\lceil \frac{len}{k} \rceil$ 次涂色才能得到这段区间。下面我们证明这一下界可以达到：随便选择一个长度不小于 k 的相同颜色的连续区间，作为最后涂色的区间。倒过来考虑，如果已知下一次涂色**最终留下颜色**的区间，那么本次涂色最终留下颜色的区间可以与下一次涂色的区间相邻，且长度不超过 k，因此每一个长度为 len 的相同颜色连续区间，都可以恰好经过 $\lceil \frac{len}{k} \rceil$ 次涂色得到。

因此将每个区间的贡献简单相加即可得到答案，总时间复杂度 $O(n)$。

I. It Takes Two of Two

作者 fstqwq　**一血** 87(+1) 上海交通大学 撸串的喵

题解　每个合作即一条边，在没有重边和自环的情况下度数最大为 2，意味着图里只会存在孤立点、链和环。

每当一条边加到：

- 两个孤立点时：形成一条长度为 1 的链；
- 一个孤立点和一条链的端点时：形成一条长度增加 1 的链；
- 一条长度大于 1 的链的两个端点时：形成一个环；
- 两条不同的链的两个端点时：形成一条长链；
- 其余情况：不可行。

考虑使用记忆化搜索实现的动态规划，记录孤立点的数量、长度为 1 的链的数量和长度大于 1 的链的数量即可，采取上述讨论转移。总复杂度 $O(n^3)$。

J. Just Some Bad Memory

作者 desprado2, fstqwq　**一血** 142(+2) 上海交通大学 梦想天生

题解　一个图没有奇环当且仅当是一个二分图。我们可以通过判断是否是二分图的方式来判断是否有奇环。

一个图没有偶环时，每条边属于至多一个环，也即是仙人掌。所有的奇环都没有公共边，否则通过删掉公共边一定能得到一个偶环。

因此，我们可以通过判断二分图来判定奇环的存在性，判断是否是仙人掌、并且仙人掌上的环均为奇环来判断偶环的存在性。第一部分可以用 DFS 染色或者并查集完成，第二部分可以用 Tarjan 打标记完成。

接下来进行巨大多讨论：

- 如果 $n < 4$，无解，否则有解。
- 如果既有奇环也有偶环，那么答案是 0，否则答案至少是 1：
  - 如果有偶环，那么答案是 1，在偶环上构造奇环即可。
  - 如果有大于 3 的奇环，或者存在奇环的同时存在一条有 4 个点的链，那么答案是 1：直接构造偶环即可。
- 否则答案至少为 2，因为此时要么既没有奇环也没有偶环（此时为森林，则加一条边至多加出一个环），要么只有大小为 3 的奇环且没有 4 个点的链（无法立刻构造出偶环）。
  - 如果有奇环，直接在奇环上接两条边即可，此时答案为 2；
  - 否则既没有奇环也没有偶环，如果存在有 4 个点的链，那么加两条边可以分别构造出奇环和偶环，此时答案为 2；
  - 否则所有链长度不超过 3。如果构造奇环的同时能够构造出有 4 个点的链，那么答案也是 2。唯一满足这个条件的情况是，存在一个度数不小于 3 的点 w，我们通过联通其两个出度 x, y 得到奇环 $\{w, x, y\}$，随后连接出度 y, z 得到偶环 $\{w, x, y, z\}$。事实上，该种情况和上种情况可以合并为存在一个大小至少为 4 的连通块。
- 否则答案不小于 3。注意到我们其实最多只需要 5 条边，因此我们仅需利用原图的至多 2 条边，而这 2 条边关系如何是任意的，因此答案为 $5 - \min(m, 2)$。

K. Known as the Fruit Brother

作者 desprado2, fstqwq　**一血** N/A

题解　考虑建图跑最短路，其中矩形的四角、爆炸果实、起点和终点是关键点。我们可以把直接走到的点连起来，把爆炸果实连起来，然后我们考虑跳跃的时候的情况：

- 如果距离足够，那么可以直接跳到；
- 如果距离不够，但是落点附近没有障碍物，并且落点到想去的点中间没有障碍物，那么可以减少 R 的距离；
- 否则，落点在障碍物中。此时，如果我们想从爆炸果实直接到下一个关键点，那么一定是跳 R 到一个障碍物边缘，也即圆与障碍物的交上。注意，这种交点只会有出度直接到达某个关键点，不会从其他点走到。如果把他们直接当做关键点的话，建图复杂度会从 $O(n^4)$ 上升至 $O(n^5)$，这样会无法通过。

总共 $O(n^3)$ 条边，每条边连边时复杂度为 $O(n)$；最短路图 Floyd 处理的复杂度是 $O(n^3)$ 。总复杂度 $O(n^4)$。

L. Last Warning of the Competition Finance Officer

作者 AntiLeaf　**一血** 204(+1) 上海交通大学 风乎舞雩

题解　不难发现可以令 f_i 表示长为 i 的前缀的答案，考虑如何转移。

假设 s_i 结尾的位置可以匹配到若干个长为 k、典值为 w 的串，那么显然有

$$f_i = \sum f_{i-k} w$$

。

匹配如果直接枚举每个 t_i 肯定会 TLE，因此需要优化。不难发现，不同的长度数量最多只有 $O(\sqrt{\sum |T_i|})$ 个。因此在 DP 的基础上有两种做法：

1. 哈希

   直接将每一个串求哈希值后放进一个表中，查表转移。另外不要用 10^9 级别模数的哈希，太长的串也不能用自然溢出，很容易卡掉。

2. AC 自动机暴力

   对所有 t_i 建 AC 自动机，并对 AC 自动机的每个结点求出它的 fail 树的祖先中最近的一个是结尾位置的结点。考虑 s 的每一位时，用这个信息一步步往上跳就可以得到所有匹配的串。

两种做法复杂度都是 $O(|S|\sqrt{\sum |T_i|})$，不过哈希的复杂度是一定会跑满的，并且因为要做 $O(|S|\sqrt{\sum |T_i|})$ 次哈希运算，常数比较大。AC 自动机做法常数非常小，大概只需要八分之一时限的时间。

M. My University Is Better Than Yours

作者 desprado2　**一血** 25(+) 复旦大学 刻耳柏洛斯

题解　考虑建图，对于每一个大学排名，从前往后扫描，将前一个学校往后一个学校连边，最后形成一个有向图。则题意转化为，求出有向图中的每一个节点可以到达多少个其他节点。

由于这个有向图存在哈密顿路（每一个大学排名都是一条哈密顿路），所以将该有向图缩强连通分量后，形成的 DAG 是一条链。基于这个性质，每个节点的答案，等于其所在强连通分量和该强连通分量在链上的所有后继的节点总数。此外，由于本题存在很多特殊性质，存在很多做法，且即使不了解强连通分量的知识也可以通过本题。几个典型的做法如下:

方法 1： 直接建图，使用 Tarjan 算法求出强连通分量，缩点后进行一轮拓扑排序即可得出链结构。总时间复杂度 $O(nm)$。

方法 2： 随便选一个大学排名作为基础排名。对于每个点，用一个并查集维护其在基础排名中最可以到达的最靠前的位置。扫描其余的大学排名，对每个排名从前往后扫描，将前一个学校的集合合并到后一个学校所在的集合中。使用路径压缩（注意此处无法按秩合并），总时间复杂度 $O(nm \log n)$。

方法 3： 在每个排名中，答案都是单调不增的，而当答案变化时，我们可以发现从这里划分开来，每个排名在这之前的标号集合和之后的标号集合完全相同。因此我们可以通过维护所有排名前 i 个中出现了 a_i 个不同的标号，当 $a_i = i$ 时，给所有现在出现过的标号中未计算答案的答案记为现在未计算答案的标号个数即可。总时间复杂度 $O(nm)$。

N. Nine Is Greater Than Ten

作者 fstqwq　**一血** 1(+) 上海大学 江海寄余生

题解　比较字符串字典序即可。C，C++，Java，Python，Rust，JavaScript，C#，PHP 等若干语言的默认字符串比较可以通过该题。