

北京理工大学第17届程序设计竞赛(决赛) - 题解

A. 赛前须知

答案：ACACACACAC

解析

1. 根据赛前须知，B、C项属于“比赛开始之前拆封试题袋、触碰鼠标或者键盘”，D项属于“使用自己的电子设备”，只有A项没有被禁止。因此选择A项。
2. 根据赛前须知，A项属于“使用自己的电子设备”，B项属于“与队友和工作人员以外的人交流”，D项属于“与除完成比赛必要的计算机以外的计算机进行网络通信”，均错误。C项属于“查阅本队在赛前准备的纸质材料”，这是被允许的。因此选择C项。
3. 根据赛前须知，本次比赛采用ICPC赛制。因此选择A项。
4. 根据ICPC赛制的规则，通过题数多的排名高，通过题数相等的情况下罚时少的排名高。因此选择C项。
5. 该队共通过3题。其中A题在第20分钟通过，罚时为20分钟；B题在第40分钟通过，且在第10分钟有一次WA的提交，罚时为 $40 + 20 = 60$ 分钟，注意根据规则，第30分钟的CE提交不计入罚时；C题在第60分钟通过，罚时为60分钟。总罚时为140分钟。因此选择A项。
6. 根据赛前须知，“本次比赛允许的竞赛语言包括：C, C++, Java, Python”，其中不包括JavaScript。因此选择C项。
7. 根据赛前须知，“如果对题目有疑问，应当通过比赛系统的“clarification”功能向裁判组提出”。因此选择A项。
8. 根据赛前须知，“当遇到不会做的题目时，请继续阅读下一道题目”。因此选择C项。
9. 根据赛前须知，本次比赛由计算机学院与信创学院联合主办。因此选择A项。
10. 根据赛前须知，“开源鸿蒙”被称为“OpenHarmony”，可以由此推断或根据常识得出“鸿蒙”的英文名为“Harmony”。因此选择C项。

B. 小L的围棋

题意简述

给 n 个嵌套的区间和一个序列，每个区间的权值是该序列上所有数的最小值。求按题目描述所给的双方的积分。

题解

嵌套的区间形成一颗树，先通过给出的区间数据建出树，树上的每个节点有两种颜色。

规则一

若子树中与该节点颜色相同的所有节点中众数个数大于另一种颜色的节点的众数个数，则计一分，可用dsu on tree处理信息，维护子树中两种颜色棋子的区间众数，由于每个区间的权值只跟元素的相对大小有关，则先对权值进行离散化，总复杂度 $O(n \log n)$ ；

规则二

由于各个节点的权值是区间最小值，整个子树中的点都是子树根节点的子区间，则对于从子树根出发往下的每一条路径都是单调不减的，对每个点求两种颜色的深度，设 $\text{dep1}[u]$ 为白色深度， $\text{dep2}[u]$ 为黑色深度， v 为 u 的子节点：

- 若该点为白子： $\text{dep1}[u]=\max(\text{dep1}[v])+1$, $\text{dep2}[u]=\max(\text{dep2}[v])$, $\text{ans}[0] += \text{dep1}[u] > \text{dep2}[u]$;
 - 若该点为黑子： $\text{dep1}[u]=\max(\text{dep1}[v])$, $\text{dep2}[u]=\max(\text{dep2}[v])+1$, $\text{ans}[1] += \text{dep1}[u] < \text{dep2}[u]$;
- 复杂度 $O(n)$;

综上所述，复杂度为 $O(n \log n)$ ；另外，离线处理众数问题和区间最长嵌套也可以用莫队，复杂度 $O(n\sqrt{n})$ ，本题数据并没有卡此种做法。

C. 小L的旅行

考虑对权值重新建个图。具体来说，如果 $i \& j = 2^k$, k 为非负整数，而且 $i > j$ ，则从 i 往 j 连一条长度为 0 的边。然后对于原图上的每个点来说，从 i 连一条到权值图上 $a[i]$ 的边，长度为 1，再从权值图上的 $a[i]$ 往原图的 i 连一条长度为 0 的边。

然后从 1 号节点跑最短路即可，通过传送门的操作就可以看成是花 1 分钟进入权值图，然后瞬移到某个节点后回到原图。

由于边权只有 01，所以建议使用 01BFS。设 $\max\{a_i\} = M$ ，则边数是 $O(M \log M)$ 的，因此复杂度是 $O(M \log M)$ 。常数小的 dijkstra 也是能通过的。

D. 小L的数学题

转换成计算所有不包含平方因子的因子个数，可以发现这和原题是等价的。

考虑容斥，钦定一个平方数，对于所有这个平方数的倍数，计算他们的倍数个数之和，容易发现容斥系数是 μ 。

因此答案就是 $\sum_{i=1}^{\sqrt{n}} \mu(i) \sum_{j=1}^{\lfloor \frac{n}{i^2} \rfloor} \lfloor \frac{n}{i^2 j} \rfloor$ 。复杂度 $O(\sqrt{n} \ln n)$

这个题还告诉我们一个非常神奇的等式，设 $g(n)$ 表示 n 的不同的质因子个数，那么：

$$\sum_{i=1}^n i \varphi(\lfloor \frac{n}{i} \rfloor) = \sum_{i=1}^{\sqrt{n}} \mu(i) \sum_{j=1}^{\lfloor \frac{n}{i^2} \rfloor} \lfloor \frac{n}{i^2 j} \rfloor = \sum_{i=1}^n 2^{g(n)} = \sum_{i=1}^n \sum_{d|n} \mu^2(d) = \sum_{i=1}^n \mu^2(i) \lfloor \frac{n}{i} \rfloor$$

E. MyGO!!!!

所求划分段数三次方可进行如下转化：

$$\begin{aligned} n^3 &= (n-1)^3 + 3(n-1)^2 + 3(n-1) + 1 \\ n^2 &= (n-1)^2 + 2n + 1 \\ n &= n - 1 + 1 \end{aligned}$$

考虑增量法，每新加入一个位置 i ，只需求出新增的划分方案数，便可由上式递推得出当前的答案。

为求得方案数，需要找出所有满足 $j \in [0, i) \wedge \bigoplus_{p=j+1}^i a_p > k$, $[1, i]$ 的划分方案数即为所有 $[1, j]$ 的方案数累加。

过程中需维护每个位置划分段数立方、平方、段数和及方案数。

设数列 a 的前缀异或为 b ，问题转化为求所有 $b_j \oplus b_i > k$ ，这个问题可以使用 01trie 解决。

具体而言，先对于 trie 的每个节点维护子树和。

对于二进制下的每一位，如果 k 当前位为 0，则可以累加和 b_i 当前位相同的儿子的贡献，并走向与当前位相反的儿子。

反之则只能走向与当前位相反的儿子，不累加贡献。

F. Harmini

不妨以 1 号点为根。

首先预处理出一个树上前缀异或和数组 s ， s_i 代表 i 号节点到 1 号点的路径上的边权的异或和。只要求出 s 数组即可算出答案。

容易发现对于一组询问 $[u, v]$ ，其结果就是 $s_u \oplus s_v$ ，又有 $s_1 = 0$ ，因此如果将一次询问视为给 s_u, s_v 连一条边，那么当所有点都联通时就可以求出 s 数组。事实上，当这些点都联通时，只要找到一颗生成树就可以让询问次数变为 $n - 1$ 次。

此时问题转化为仅用合法的询问 (即在原树中距离恰为 k 的点对)，能否让所有的 s_i 联通，并找出任意一颗生成树。所有距离为 k 的点对数量是 $O(n^2)$ 的，因此不能暴力求。

将所有距离为 k 的点对 u, v 分为两类：

- u, v 之间有父子关系
一次 dfs，将当前点与 dfs 栈中倒数第 $k + 1$ 个点连接。
- u, v 之间没有父子关系

dfs 时，对每个点维护一个 `map<int, vector<int>> mp`，`mp[i]` 代表当前点的子树中深度为 i 的点的集合。合并子树时采用启发式合并，即较小的容器向较大的容器合并。该操作是 $n * \log^2(n)$ 的。

在合并两颗子树时，设当前节点深度为 d ，枚举较小的子树内的节点 v ，则其与另一棵子树内的所有深度为 $deep[u] = k + 2 * d - deep[v]$ 的节点 u 距离为 k ，此时应当连接它们。这样会让所有合法点对在他们的 lca 处被计入，可以做到不重不漏，因此暴力连接依然会连接所有的距离为 k 的点对，有至少 $O(n^2)$ 次操作，考虑如何简化这个过程。

如果同一个深度内的一些点已经在连通块内，则只需要保留一个点。因此当枚举到 v 时，可以先将所有 $deep[u] = k + 2 * d - deep[v]$ 的点 u 连接，然后仅保留一个，与 v 节点同一深度的节点同理。这样的复杂度会退化回 $o(n * \log^2(n))$ ，证明如下：

每次操作后，一个子树被操作过的深度内至多只会保留一个节点

- 仅考虑合并的复杂度时，不超过未优化时的复杂度 $o(n * \log^2(n))$
- 考虑连边的复杂度
 - 对于被保留的节点之间，单次操作仅会连 1 条边，总次数为 $o(n * \log(n))$ 。
 - 对于没被保留的节点，其只会在被删除的时候连一次边，总次数为 $o(n)$ 。

因此总复杂度为 $o(n * \log^2(n))$

至此，与包含所有合法的边的等价的图就被处理出来了，当它联通时，按任意一颗生成树询问 $n - 1$ 次就可以求出 s 。

事实上上面的操作使用并查集维护会好写得多，merge 时顺便询问就可以了。

G. 小P玩钢4

考虑两对相邻的有敌人的格子，可以发现无论先攻击哪一个都会多消耗 1 个单位的装备。所以最后的花费和进攻顺序是没有关系的。

因此答案就是敌人总数量加上有多少对敌人是相邻的。

H. LCA Determinant

本题定位智慧 / 猜结论 / 线代高手题，结论为

$$\det A = a_1 \prod_{\substack{u \in V \\ u \neq 1}} (a_u - a_{\text{fa}(u)})$$

其中 V 是全体树上节点构成的集合， $\text{fa}(u)$ 代表节点 u 在树上的父亲节点。

赛时技巧而言，样例已给出树为单链之一例，选手应考虑补充举例菊花图的情况，并分别应用行列式消法变换证明两种特殊情况的正确性。分析两种消去方法的共性便能得到上述结论的雏形。

一般地，考虑对节点按深度从小到大进行重标号，由于重标号对行和列进行的置换相同，行列式的值和正负性均不会发生变化。现在归纳地证明结论。当 $n = 1$ 时，结论平凡。当 $n \geq 2$ ，假设结论对全体大小为 $n - 1$ 的树成立。现给定一颗大小为 n 的按上述方法重标号后的树，设其对应行列式为 A_n 。由重标号方法，编号为 n 的节点一定为叶子节点；且由 $n \geq 2$ ，其父节点存在。在行列式中做消法变换：

- 将第 n 列减去第 $\text{fa}(u)$ 列；
- 将第 n 行减去第 $\text{fa}(u)$ 行。

行列式变为

$$\begin{aligned} & \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & a_{\text{lca}(1,n)} \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & a_{\text{lca}(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{lca}(n,1)} & a_{\text{lca}(n,2)} & \cdots & a_{\text{lca}(n,n)} \end{pmatrix} \\ \rightarrow & \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & a_{\text{lca}(1,n)} - a_{\text{lca}(1,\text{fa}(n))} \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & a_{\text{lca}(2,n)} - a_{\text{lca}(2,\text{fa}(n))} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{lca}(n,1)} & a_{\text{lca}(n,2)} & \cdots & a_{\text{lca}(n,n)} - a_{\text{lca}(n,\text{fa}(n))} \end{pmatrix} \\ = & \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & 0 \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{lca}(n,1)} & a_{\text{lca}(n,2)} & \cdots & a_n - a_{\text{fa}(n)} \end{pmatrix} \\ \rightarrow & \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & 0 \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{lca}(n,1)} - a_{\text{lca}(\text{fa}(n),1)} & a_{\text{lca}(n,2)} - a_{\text{lca}(\text{fa}(n),2)} & \cdots & a_n - a_{\text{fa}(n)} \end{pmatrix} \\ = & \begin{pmatrix} a_{\text{lca}(1,1)} & a_{\text{lca}(1,2)} & \cdots & 0 \\ a_{\text{lca}(2,1)} & a_{\text{lca}(2,2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n - a_{\text{fa}(n)} \end{pmatrix} \\ = & (a_n - a_{\text{fa}(n)}) \det A_{n-1} \\ = & a_1 \prod_{\substack{u \in V \\ u \neq 1}} (a_u - a_{\text{fa}(u)}) \end{aligned}$$

故结论对任意 $n \in \mathbb{N}_+$ 均成立.

回顾证明, 我们对行和列进行对称的消法变换, 最终将矩阵化为对角矩阵, 从而计算出行列式的值. 按照线性代数的观点, 若将这些变换用矩阵 P 记录, 上述过程事实上完成了对称矩阵 A 的合同对角化, 即 $P^T A P = \Lambda$. 我们指出, 用于合同对角化的矩阵 P 完全由树上祖先关系决定, 且 $\det P = 1$ (请读者练习). 一般地, 类似结论对任意下确界存在的偏序集成立. 感兴趣的同学可参考

- Wilf, H. S. (1968). Hadamard determinants Möbius functions, and the chromatic number of a graph. *Bulletin of the American Mathematical Society*, 74(5), 960-964.

I. 星光指引前路

给定 n 个带权矩形问 m 个点每个点 i 被覆盖的所有矩形中第 k_i 小权值

整体二分+树套树 $\log^3$ 应该是过不去的 代码量常数巨大

好像能用 $\log^2$ 的奇怪分治树套树过 出题人只讲自己的方法

sol: KD树+根号平衡

K-D Tree是一种能维护高维空间的数据结构 维护 k 维空间时其单次操作时间复杂度为 $O(n^{\frac{k-1}{k}})$

教学: <https://www.luogu.com.cn/blog/van/qian-tan-pian-xu-wen-ti-yu-k-d-tree>

本题 $k=2$ 离线下来将 m 个询问点建树

n 个矩形放到KD树上, 总共会落到 $O(n\sqrt{m})$ 个节点上

将矩形的权值挂在这些节点上(vector) 分为下传的vector和自己的vector

从根开始依次向下访问左右孩子(dfs前序遍历)

每访问一个节点 取其下传vector插入到一个数据结构里

这个数据结构能查当前所存储元素中的第 k 小

存储的元素即是覆盖当前节点的矩形的权值

然后将其自己的vector也插入进去 查询记录答案后删除

随后递归儿子 回溯时删除下传vector

至于能查 k 小的数据结构 若使用平衡树类 修改查询都是 $\log$

而修改总次数是 $O(n\sqrt{m})$ 查询次数是 m 总复杂度 $O(n\sqrt{m} \log n)$

若使用根号平衡的值域分块 下标为权值统计每种矩形权值的出现次数 维护块内和

这样可以在单次 $O(\sqrt{n})$ 的时间内查询 即从小到大枚举块 就能找到第 k 小应该在哪个块内

然后在块内一个个找即可 插入单次 $O(1)$ 即单点++, 块和++ 就保证了总复杂度为 $O(n\sqrt{m} + m\sqrt{n})$

J. pass

根据题目描述, 车重量集中在最低下, 所以不会翻车

讨论一些情况即可通过:

1. 路面宽度 $>$ 车宽, 直接过, 底盘0即可。
2. 路面宽度 $\leq$ 车宽, 需要借助墙壁。
 1. 墙壁高度 $\geq$ 车宽, 无法借助墙壁, 输出-1。

2. 墙壁高度 $<$ 车宽，此时右侧轮胎可以在墙壁上。此时想要达到最小的底盘高度显然是使得左轮胎贴在路边，通过 $l = \sqrt{w * w - b * b}$ 即可得到倾斜在墙壁上的车在地面的投影长度。

1. 若此时 $l \leq a$ ，说明当车的右轮放在墙壁边缘的时候，左轮还有地方放，因此答案是0。

2. 否则，将左轮放置在路面的最左侧，右轮放置在墙壁距离端点 $l - a$ 处，通过相似比即可算出此时底盘的高度为 $\frac{(l-a)*b}{w}$ 。

K. str进制

题意简述

给一个十进制数，求它在某个进位规则下进位所表示出的数

题解

从低位到高位逐位模以进制基数，再除以进制基数，记录值后输出即可，复杂度 $O(m)$ ，注意要开long long。

L. Equation

对方程变形，得到 $n(n - 1) \equiv 0 \pmod{M}$ 。也就是说， n 和 $n - 1$ 覆盖了 M 的所有因子。

注意到 n 和 $n - 1$ 是互质的，即二者的质因子集合没有包含，同时 M 至多有 8 个不同的质因子。因此可以考虑枚举 M 的质因子是如何分布在 n 和 $n - 1$ 中的。

不妨假设 $n = ax, n - 1 = by$ ，其中 $ab = M, \gcd(a, b) = 1$ ，则有 $ax - by = 1$ 。该方程可以用扩展欧几里得算法(exGCD)求解得到 x, y 的通解。容易发现满足 $0 \leq ax < M$ 的解有且仅有一个，将其加入答案集合即可。

设 M 有 k 个不同的质因子，则可以枚举得到 2^k 个方程，每个方程都能在 $\log M$ 的时间内找到恰好一个解，因此此部分最坏情况下的复杂度为 $2^8 \log M$ 。分解质因子可以选择 $O(\sqrt{M})$ 完成。单组复杂度 $O(\sqrt{M} + 256 \log M)$ ，总复杂度 $O(T(\sqrt{M} + 256 \log M))$ ，足以通过本题。

M. 小H的糖果

题目大意

给一个字符串，允许更改其中一个字母，求能更改出的字典序最大的后缀

题解

观察题目易知把某个非z的字符更改为z一定是最优的策略，对于该串的后缀来说，尝试更改它每个后缀中第一个非Z的元素再与当前最大值比较即可，时间复杂度 $O(n^2)$