第七届“帆软杯”东南大学大学生程序设计竞赛

专业组

正式题解

计拔 2025 级　LgxCute

首先，感谢朋友们的参与！

本次专业组的题目相对基本。ChatGPT 5 Thinking 能稳定解决 12 道题中的 11 道。

注意题目不是按难度排列的。

A　宠物店 (easiest)

全局、每行和每列各维护一个标记，初始时均为 0。全局加 k 时，全局的标记加 k；第 x 行加 k 时，第 x 行的标记加 k；第 y 行加 k 时，第 y 行的标记加 k。

那么，第 x 行第 y 列的数 = 全局的标记 + 第 x 行的标记 + 第 y 列的标记。

注意使用 64 位整数（如 C/C++ 中的 `long long`）。

时间复杂度为 $O(m)$。

B　广义线段树 (medium-easy)

本题的 idea 来自 shigure，由笔者进行了加强。

线段树节点 p 在区间 $[l_i, r_i]$ 的 visit 过程中被访问，首先要求 $[L_p, R_p]$ 与 $[l_i, r_i]$ 有交，即 $l_i \le R_p$ 且 $L_p \le r_i$；其次，若 p 有父节点 fa，还要求 $[L_{fa}, R_{fa}]$ 不包含于 $[l_i, r_i]$，即“$l_i \le L_{fa}$ 且 $R_{fa} \le r_i$”不成立（值得指出的是，若引号内的条件成立，则开头的条件也成立）。

对于每个线段树节点 p，在平面直角坐标系 xOy 上放一个第一类点 (L_p, R_p)，若 p 有父节点 fa 则再放一个第二类点 (L_{fa}, R_{fa})。那么，区间 $[l_i, r_i]$ 在 visit 过程中访问的节点个数 = 矩形“$1 \le x \le r_i$ 且 $l_i \le R_p \le n$”内（含边界上，下同）的第一类点的个数 − 矩形“$l_i \le x \le n$ 且 $1 \le y \le r_i$”内的第二类点的个数。这两类点各成一个二维数点问题。

对偶地，对于每个区间 $[l_i, r_i]$，在平面直角坐标系 xOy 上放一个点 (l_i, r_i)。那么，线段树节点 p 在全部 m 次 visit 的过程中被访问的次数 = 矩形“$1 \le x \le R_p$ 且 $L_p \le y \le n$”内的点的个数 − 矩形“$1 \le x \le L_{fa}$ 且 $R_{fa} \le y \le n$”内的点的个数（其中 fa 表示 p 的父节点，若不存在则忽略此项）。这是一个二维数点问题。

离线下来进行扫描线，使用树状数组支持单点加、区间求和即可。

时间复杂度为 $O\big((n + m) \log n\big)$。

C 地球上最后的夜晚 (medium)

本题的 idea 来自 sjcx，笔者给出了证明。

不妨考虑删除哪些边。方便起见，我们重新叙述题意，以后都基于此进行讨论：

转化后的题意 选择若干条边，作为原题中删除的边（即原题中不选的边）。对于原图中的偶度点，要求有奇数条以它为端点的边被选，以后称之为**奇点**；对于原图中的奇度点，要求有偶数条以它为端点的边被选，以后称之为**偶点**。在满足上述要求的前提下，要求选择的边的编号尽可能大（严谨地说，选择的边中编号第一小的边的编号尽可能大，在此前提下选择的边中编号第二小的边的编号尽可能大，以此类推，注意不存在视作无穷大）。

首先，对于原图中的每个连通块，其内部的奇度点个数必须是偶数，否则无解。

我们断言，只要满足这一条件，便有解。对于每个连通块，考虑取其任意一棵生成树，把能选择的边限定在树边中。容易发现：

引理 1 对于任意一棵树，任意规定其上偶数个节点为奇点，其余节点为偶点，都存在且仅存在**唯一的选边方案**，满足奇点和偶点的要求。

证明 一方面，如果一条边两侧的子树各有奇数个偶点，那么这条边必须被选，因为不选相当于把问题分成两个无解的子问题。另一方面，选择全部满足上述条件的边是一个合法的选边方案，因为每个奇点有奇数个邻接子树包含奇数个奇点，每个偶点有偶数个邻接子树包含奇数个奇点，所以奇点有奇数条邻边被选，偶点有偶数条邻边被选。

可见，上述选边方案是*唯一的极小选边方案*，这对本题来说其实足够了。更进一步地，下证它是唯一的选边方案：假设存在另一个选边方案，其必然是在极小选边方案的基础上多选了若干条边。这些新选的边的导出子图是由若干棵节点数 ≥ 2 的树构成的森林，其中存在度数 $= 1$ 的节点。这样的节点在原来的树中显然不满足奇点和偶点的要求，与假设矛盾。■

由此可知，断言成立。同时，不难想到：

引理 2 对于每个连通块，把边按编号从大到小排序后进行 Kruskal 算法，得到一棵生成树，其唯一的选边方案即为所求。

证明 设得到的生成树所包含的边的编号从小到大依次是 $id_1, id_2, \cdots, id_k$。

归纳地，假设已证在编号 $\leq id_{i-1}$（$1 \leq i \leq k$；特别地，规定 $id_0 = -\infty$）的边中应选 $id_1, id_2, \cdots, id_{i-1}$。首先，不选编号在 (id_{i-1}, id_i) 内的边仍然有解，故不选。其次，若不选编号在 (id_{i-1}, id_i) 内的边，则编号为 id_i 的边必须被选，因为由 Kruskal 算法的过程知去除编号更小的不选的边后它是一条割边，由引理 1 的构造方法知它两侧的连通分量各有奇数个奇点，不选相当于把问题分成两个无解的子问题。因此，编号 $\leq id_i$ 的边中应选 $id_1, id_2, \cdots, id_i$。

最后，由 Kruskal 算法的过程知 id_k 就是连通块中编号最大的边的编号。■

综上所述，对于每个连通块，把边按编号从大到小排序后进行 Kruskal 算法，得到一棵生成树，用引理 1 的构造方法得到其唯一的选边方案，这里选择的边就是原题中不选的边。

时间复杂度为 $O\big(n + m \cdot \alpha(n)\big)$。

Remark 根据断言，按编号从小到大依次判断每条边能否不删除也是可行的，但一般需要使用复杂度稍劣或常数巨大的数据结构，实现复杂且难以通过本题。

D 孤独之歌 (medium-easy)

本题的 idea 来自 sjcx。

考虑在末尾添加一个字符时暂不区分它是操作 `0` 还是操作 `1`。对于任意一个操作序列，每个添加操作所对应的字符是否被保留到最后是确定的。若被保留到最后，则它是操作 `0` 还是操作 `1` 也是确定的；若未被保留到最后，则它既可以是操作 `0` 也可以是操作 `1`，不妨在删除它的操作 `B` 处再区分。

基于这样的想法，考虑动态规划，记 $f[i][j]$ 表示用 i 次操作打出一个确定的长度为 j 的字符串的方案数。初始时，$f[0][0]=1$。采用“刷表法”，$f[i][j]$ 的转移如下：

- 在末尾添加一个字符：加给 $f[i+1][j+1]$。
- 操作 `B`：
  - 若 $j>0$，乘 2 加给 $f[i+1][j-1]$。
  - 若 $j=0$，加给 $f[i+1][0]$。

时间复杂度为 $O(n|S|)$。

Bonus (medium-hard) ChatGPT 指出，本题还存在 $O(n)$ 的算法。

记 op_i（$1\le i\le |S|$）表示所对应的字符被保留到最后作为 S_i 的添加操作。

在 op_i 与 op_{i+1}（$i=1,2,\cdots,|S|-1$）之间、$op_{|S|}$ 之后，可以插入一个“括号序列”（以添加操作为“左括号”，操作 `B` 为“右括号”，每对“括号”使系数乘 2）。设其普通生成函数为 $G(x)$，x^k 的系数表示长度为 k 的“括号序列”的系数之和，有

$$G(x)=2x^2G(x)^2+1$$

（第一个“左括号”与它所对应的“右括号”之间、“右括号”之后各是一个“括号序列”；注意此处的“括号序列”可以是空的，还需要补上 x^0 的系数 1）

在 op_1 之前，可以插入任意多（可以是 0）个“括号序列”和操作 `B`。设其普通生成函数为 $H(x)$，x^k 的系数表示总长度为 k 的“括号序列”和操作 `B` 的拼接的系数之和，有

$$H(x)=\big(2x^2G(x)+x\big)H(x)+1\Longrightarrow H(x)=\frac{1}{1-x-2x^2G(x)}$$

（考虑第一部分是一个“简单括号序列”还是一个操作 `B`；此处的“简单括号序列”指无法被划分成多个“括号序列”的“括号序列”，这样规定是为了避免重复计数）

那么，答案 $ans=[x^n]x^{|S|}G(x)^{|S|}H(x)=[x^n]\dfrac{x^{|S|}G(x)^{|S|}}{1-x-2x^2G(x)}$。

由 $G(x)=2x^2G(x)^2+1\Longrightarrow 1=2x^2G(x)+\dfrac{1}{G(x)}\Longrightarrow 1-x-2x^2G(x)=\dfrac{1-xG(x)}{G(x)}$ 得

$$ans=[x^n]\frac{x^{|S|}G(x)^{|S|+1}}{1-xG(x)}=[x^{n+1}]\frac{\big(xG(x)\big)^{|S|+1}}{1-xG(x)}=[x^{n+1}]\sum_{k=|S|+1}^{\infty}\big(xG(x)\big)^k$$

记 $F(x)=xG(x)$，$\Phi(u)=2u^2+1$，有 $F(x)=x\Phi\big(F(x)\big)$，由 Lagrange 反演得

$$[x^m]F(x)=\frac{k}{m}[u^{m-k}]\Phi(u)^m=\begin{cases}\frac{k}{m}\binom{m}{\frac{m-k}{2}}2^{\frac{n-k}{2}} & n-k\text{ 是偶数}\\ 0 & n-k\text{ 是奇数}\end{cases}$$

因此，$ans=\sum\limits_{k=|S|+1}^{\infty}[x^{n+1}]F(x)^k=\sum\limits_{i=0}^{\left\lfloor\frac{n-|S|}{2}\right\rfloor}\frac{n+1-2i}{n+1}\binom{n+1}{i}2^i$。

E 可靠的猪宝宝 (medium-easy)

这是一道典型的数位统计类动态规划题。首先，$[L, R]$ 的答案 $=[1, R]$ 的答案 $-[1, L-1]$ 的答案。下面，我们讲述怎么求 $[1, N]$ 的答案.

暂时忽略额外的要求，考虑怎么求 $\sum\limits_{x=1}^{N} 2025^x$。下称十进制表示中位权为 10^i 的位为第 i 位，设 N 的最高位为第 m 位，N 的第 i（$0 \le i \le m$）位是 N_i。记 $f[i][0/1]$ 表示“只考虑第 $i \sim m$ 位（这些位全部相同的只计一次），这些位是否（是对应 1，否对应 0）全部与 N 相同”的 2025^x 之和。初始时，$f[m+1][1]=1$。采用“刷表法”，转移如下：

- 对于 $f[i][0]$，枚举 $d=0,1,\cdots,9$，乘 $2025^{10^{i-1}d}$ 加给 $f[i-1][0]$。
- 对于 $f[i][1]$：
  - 枚举 $d=0,1,\cdots,N_{i-1}-1$，乘 $2025^{10^{i-1}d}$ 加给 $f[i-1][0]$。
  - 对于 $d=N_{i-1}$，乘 $2025^{10^{i-1}d}$ 加给 $f[i-1][1]$。

这里用到的 $2025^{10^{i-1}d}$ 可以 $O(m \cdot base)$ 地递推出来，其中 $base=10$。

最后，$\sum\limits_{x=1}^{N} 2025^x = f[0][0]+f[0][1]$。

对于额外的要求，只需额外增加状态维度，以把计数对象进一步分类。通过状态，要能判定计数对象是否满足要求。状态及其之间的转移，本质上构成确定性有限状态自动机。

对于“$2,3,5,7$ 都不是 x 的约数”，直接的想法是记录模 $2,3,5,7$ 的结果，但这样的状态数稍多。容易想到，是不是 2 和 5 的约数只与个位数有关，因此只需记录模 3 和 7 的结果。

对于“x 对应的数字串不存在子串 2025”，只需记录 2025 的“KMP 自动机”（即单模式串的 Aho-Corasick 自动机）的状态，规定不允许转移到表示匹配成功的状态即可。

时间复杂度为 $O(|Q| \cdot base \cdot \log R)$，其中 $|Q| = 3 \times 7 \times 4 = 84$。

F XCPC 重开 (medium-easy)

我们先说明一个基本事实：

基本事实 存在最优策略，满足：做数字编号为 i 的题花费恰好 t_i 分钟；比赛中做题的时间是一个前缀；做同一道题的时间是连续的；在要做的题中，先做 t_i 小的，再做 t_i 大的。

证明 下证任意不满足上述条件的策略都可以被调整成满足上述条件且不劣的策略：

首先，若做数字编号为 i 的题花费小于 t_i 分钟，由于未通过，还不如不做；若花费大于 t_i 分钟，由于花费 t_i 分钟就能通过，超出 t_i 分钟的时间可以节省下来。

其次，若比赛中做题的时间不是一个前缀，则把做题的时间向前调整更优。

再次，设通过的题目按通过时间从小到大排序后数字编号依次是 $p_1, p_2, \cdots, p_m$，则通过数字编号为 p_k（$k = 1, 2, \cdots, m$）的题的时间不早于第 $\sum\limits_{i=1}^{k} t_{p_i}$ 分钟，采用“做同一道题的时间连续”的策略时都取到最小值。因此，罚时最少是 $\sum\limits_{i=1}^{m}(m-i+1)t_{p_i} + \sum\limits_{i=1}^{m} d_{p_i}$。

最后，若 $p_1, p_2, \cdots, p_m$ 不是按 t_{p_i} 从小到大排序的，则存在 i, j 满足 $i < j$ 且 $t_{p_i} > t_{p_j}$。因为 $(m-i+1)t_{p_i} + (m-j+1)t_{p_j} > (m-i+1)t_{p_j} + (m-j+1)t_{p_i}$，所以交换 p_i 和 p_j 更优。又因为每次交换都会使这样的逆序对减少，所以经过有限次交换调整便会完成。■

基于此，先把题目按 t_i 从小到大排序。考虑动态规划，记 $f[i][j]$ 表示“考虑排序后的前 i 道题，做题花费 j 分钟”的最优结果。这里的结果包含两个关键字——第一关键字是通过的题目数量，第二关键字是罚时。采用“刷表法”，分做和不做两种情况转移即可。

为了输出方案，还需记录每个状态的最优结果是从哪转移过来的，最后倒推回去。

时间复杂度为 $O(\sum n \cdot timeLim)$，其中 $timeLim = 300$。

Remark 按 t_i 排序后直接贪心是错误的。t_i 较小的题目的 d_i 可能很大。

G 礼物 (easy)

记 $sum_k = \left(\sum\limits_{i=1}^{k} a_i\right) \bmod n$（$0 \le i \le n$）。它有 $n+1$ 项，取值却不超过 n 种。

由鸽巢原理，存在 x, y（$x < y$）满足 $sum_x = sum_y$，即 $\left(\sum\limits_{i=x+1}^{y} a_i\right) \bmod n = 0$。

为每种取值都开一个桶即可。

时间复杂度为 $O(n)$。

Remark 简单的随机算法一般是错误的。设 $n = 2(2k+1)$，$p = 2(k+1)$，假如序列 a 中有 $2k+1$ 个 p 和 $2k+1$ 个 1，由于 $\gcd(2k+1, k+1) = 1$，唯一的方案是选 $2k+1$ 个 p。

H 概率论 (medium-easy)

记 p_i 表示经过第 i 级台阶的概率，即“概率树”上状态为“在第 i 级台阶”的节点的概率之和。$p_0=0$，$p_1=1$。对于 $i\in\{2,3,\cdots,n-1\}$，考虑状态为“在第 i 级台阶”的节点的父节点，状态为“在第 $i-1$ 级台阶”的节点有一个概率为自身的 $\frac{1}{2}$ 的状态为“在第 i 级台阶”的子节点，状态为“在第 $i-2$ 级台阶”的节点有一个概率为自身的 $\frac{1}{2}$ 的状态为“在第 i 级台阶”的子节点，所以 $p_i=\frac{1}{2}p_{i-1}+\frac{1}{2}p_{i-2}$，/即

$$\begin{bmatrix}p_i\\p_{i-1}\end{bmatrix}=\begin{bmatrix}\frac{1}{2}&\frac{1}{2}\\1&0\end{bmatrix}\begin{bmatrix}p_{i-1}\\p_{i-2}\end{bmatrix}=\cdots=\begin{bmatrix}\frac{1}{2}&\frac{1}{2}\\1&0\end{bmatrix}^{i-1}\begin{bmatrix}1\\0\end{bmatrix}$$

考虑到 m 是十进制大整数，可以递推出 $\begin{bmatrix}\frac{1}{2}&\frac{1}{2}\\1&0\end{bmatrix}$ 在模 10^9+7 意义下的 10^i 次幂。

当然，也可以直接推导 p_i 的通项公式，这在高中数学范围内。

特别地，$p_n=p_{n-1}+\frac{1}{2}p_{n-2}$。由实际意义，$p_n=1$，因此特判 $n=m$ 的情况即可。

时间复杂度为 $O(\log n)$。

I 心意不尽，前程已近 (medium-easy)

首先，考虑“对于点集的任意一个子集，两端都在子集内的边（无论是否已定向）的条数不超过子集的大小”的含义。这一特殊性质说明，每个连通块的节点数都不超过边数，即每个连通块都是树或基环树（忽略边的方向，下同）。由于其导出子图的每个连通块也都是树或基环树，这一特殊性质其实是“每个连通块都是树或基环树”的等价表述。

处理基环树，常使用类似于拓扑排序的方法不断剥离叶节点，最后剩下环。

对于每个环，由于它成为有向环只有两种可能，容易求出它的合法定向方案数。

答案 = 各个环的合法定向方案数（无未定向边则为 1）之积 $\times\ 2^{\text{不在环上的未定向边数}}$。

时间复杂度为 $O(n)$。

J 新鲜的日子和可爱的日子 (easy)

假设 S 由长度为 $l_1,l_2,\cdots,l_k$ 的段依次构成，同一段内字符相同，相邻段间字符不同。在第 i 段中，若 $l_i\geq 2$ 则有 2 个新鲜的日子和 l_i-2 个可爱的日子，若 $l_i=1$ 则都没有。

若 m_1 是奇数，无解；否则，恰有 $\frac{m_1}{2}$ 个长度 ≥ 2 的段，暂且认为它们的长度都是 2。

若 $m_2>0$ 但 $m_1=0$，无解；否则，向长度 ≥ 2 的段中加入 m_2 个字符。

最后，加入 $n-m_1-m_2$ 个长度为 1 的段。

时间复杂度为 $O(n)$。

K 无聊的猪宝宝 (easy)

可以用两个集合的容斥原理来计算答案。

把 $a_1, a_2, \cdots, a_n$ 划分成 m 段，相当于从 $n-1$ 个邻项之间的间隙中选 $m-1$ 个插入隔板。设满足 $a_i > a_{i+1}$ 的间隙有 x 个，满足 $a_i < a_{i+1}$ 的间隙有 y 个。

要使条件 A 成立，必须且只需在满足 $a_i > a_{i+1}$ 的间隙插入隔板，方案数为 2^{n-1-x}。

要使条件 B 成立，必须且只需在满足 $a_i < a_{i+1}$ 的间隙插入隔板，方案数为 2^{n-1-y}。

要使两者同时成立，必须且只需在满足 $a_i \neq a_{i+1}$ 的间隙插入隔板，方案数为 $2^{n-1-x-y}$。

所以，答案是 $2^{n-1-x} + 2^{n-1-y} - 2^{n-1-x-y}$。

时间复杂度为 $O(n)$。

L 量子计算程序设计 (easy)

这是一道诈骗题。

一个 SWAP 门可由两个 CNOT 门拼成。先 `CNOT 1 3`，再 `CNOT 3 1` 即可。

选手也可以借助 $|\psi_2\rangle$，实现略微笨拙的算法，或在制备 Bell 态后进行隐形传态。