牛客竞赛
AC.NOWCODER.COM

2025 牛客
暑期多校训练营 10

南京外国语学校

试题难度

预期试题难度：

- Easy： [D]、[H]、[I]
- Easy-Medium： [E]、[F]、[K]
- Medium： [G]、[L]
- Medium-Hard： [A]、[B]
- Hard： [C]、[J]

AC.

通过情况

题目编号和标识	[A] spacy	[B] trim
通过提交数	1	10
首次通过时间	02:47:11	02:42:12
题目编号和标识	[C] hexagis	[D] grammar
通过提交数	15	1335
首次通过时间	02:30:46	00:04:17
题目编号和标识	[E] affection	[F] yuuka
通过提交数	400	524
首次通过时间	00:20:22	00:04:51（作弊）

通过情况

题目编号和标识	[G] starlight	[H] equation
通过提交数	30	1197
首次通过时间	00:58:36（作弊）	00:07:52
题目编号和标识	[I] matrix	[J] segments
通过提交数	906	10
首次通过时间	00:09:54	02:26:10
题目编号和标识	[K] amazing	[L] desertrium
通过提交数	224	6
首次通过时间	00:20:24	03:19:51

吐槽

[A] The Spacy Language

本题难点在于代码。考察选手理解题目以及模块化编程的能力。

值得一提的是，函数名称可以重复，也可以是 main。同时，需要将诸如 i+++i 的代码片段处理为 i++ + i。特别地，C++ 中需要谨慎处理 getline 导致吞掉换行等问题。

[B] Trimming Palindrome

看到题目中这么多关于回文的条件，第一反应是把 s 的回文树给建出来再思考。

考察该回文树的 fail 树（将 0 和 1 两个空点合并），认真分析题目中的式子与 fail 树的性质，就有了下面这个暴力 dp：$f(u,i)$ 表示现在的节点是 u，$dep=i$，能够获得最大的愉悦度。我们在 fail 树上做这个 dp，有：

$$f(u,dep(u)) = \max_{v\in subtree(u)}\{f(v,dep(v))+a_{sz_u}*dep_v-a_{sz_u}*(dep_u-1)+c_{len_u}*b_{sz_u}\}$$

直接 dp 是平方的。可以考虑使用自 fail 树叶子向上合并的李超树优化，因为常数极小所以 $\Theta(n\log n)$ 的复杂度已经足以通过本题。注意不要自根向下，这样子因为深度原因常数较大无法通过。

[B] Trimming Palindrome

能不能做到 $\Theta(n)$ 呢？考虑转化式子。我们让 u 继承儿子的 dp 值。

$f(u,d)=\max_{v\in son(u)} f(v,d)$

$f(u,dep(u))=\max_{v\in son(u)}\max_d\{f(v,d)+a_{sz_u}*d-a_{sz_u}*(dep_u-1)+c_{len_u}*b_{sz_u}\}$

考虑长链剖分优化。每个点从重儿子得到 dp 出的值，轻儿子暴力改。如何得到值？可以在每个 u 上维护一个凸包，修改的时候暴力重构前面部分的凸包，后面部分的凸包容易发现只会删不会加。而删掉了之后就会少一个节点，所以修改部分是线性的。查询时因为 a 的单调性可以发现越往后，a 越大，所以查询的时候也可以直接暴力，把不可能成为决策点的线段弹出。

这样子的复杂度容易证明是线性的。

[C] Hexagis

先把所有可能的六连块找出来，这可以通过手动枚举、搜索等多种方式实现。

为了求出朋友的解答中每种块有多少个，可以对方格表进行 dfs 以找出所有的块，如果其中有不是六连块的块就直接输出 -1，否则可以将先前找到的所有六连块存入一个表中，再枚举所有六连块，判断当前搜索到的块和哪一个同构。

接下来需要给出一种满足要求的答案。这是一个拼图问题，可以转化为精确覆盖问题。具体地，我们构造一个若干行、420 列的 01 矩阵。前 360 列表示我们需要覆盖所有的格子，后 60 列表示我们需要每种六连块用恰好一次。

[C] Hexagis

枚举所有可能的六连块以及旋转角度，再枚举将其放在哪一个位置。对于每一种位置创建一个行，将其覆盖的所有格子对应的列设为 1，并将它的六连块种类对应的列设为 1，其余列设为 0。这样满足要求的解与这个矩阵的精确覆盖构成一一对应关系。

写一个 DLX 算法暴搜，找到一种精确覆盖，然后用一些方法转化为对应的拼图即可。当然，对于这么大的一个矩阵，是不太可能在题目的时间限制内找到解的，不过可以先让程序在本地运行，找到解之后再直接把输出解的语句交付评测。验题人的搜索程序运行 30 分钟即可得到答案。

[D] Grammar Test

不妨设 $x > 0, y > 0, x \neq y$，那么经历至少一次第一类或第四类操作后，最终 x, y 中至少一个的值为 0 或者两个值相等，此时必定不合法。

故所有操作均为第二类或第三类操作，即序列 a 中任意相邻两项不同。故可能合法的 a 只有 $\{0, 1, 0, 1, ...\}$ 和 $\{1, 0, 1, 0, ...\}$ 两种。

注意到每执行三次可以交换两个元素的值。所以只要判断 n 是否是三的倍数并且商是奇数即可得知答案是 2 或 0。

[E] Sensei and Affection

解法 1:

若 $m=1$，直接贪心。注意到全体最大值永远不会变大，每次选取一个极长的最大值没达到全体最大值的段操作即可。

若 $m=2$，枚举两种数之差 δ，将数组差分，将操作转化为单点加一减一。dp，维护每个位置之前有几个没配上减一的加一，再维护每个位置的差分是 $0,-\delta,\delta$ 即可。注意 $-\delta,\delta$ 间隔出现。

复杂度 $O(nV+nV^2)$。

[E] Sensei and Affection

解法 2:

注意到将全 0 序列变为给定序列 x 的操作数是 $\frac{\sum_{i=1}^{n+1}|x_i-x_{i-1}|}{2}$。

若 $m=1$，直接贪心。注意到全体最大值永远不会变大，计算将全 0 序列变为序列 $\max-a_i$ 的操作数即可，$\max$ 是 a_i 最大值。

若 $m=2$，枚举两种数 x,y，利用注意到的公式 dp。记录当前的 a_i 变为 x 还是 y，并维护当前位置初始值与目标值之差与上一个数的初始值与目标值之差的差的绝对值，累加后输出时除以 2 即可。注意枚举的值域上界不止 100。

复杂度 $O(nV+nV^2)$。

[F] Sensei and Yuuka Going Shopping

由于一共两个断点将原数列分成三段，考虑枚举第一个断点的位置，并利用数据结构维护第二个。

记 nxt_i 表示元素 a_i 下一次出现的位置的下标。记 lst_i 表示元素 i 最后一次出现的位置的下标。

考虑枚举第一个断点（即 b_1）在 $i \to i+1$ 的过程。

首先，实时维护 S_1。若 S_1 在未添加 a_i 时就已存在 a_i，那么对于第二个断点在 $b_2 \in [i+2, nxt_i]$ 的位置的答案减 1。否则，对于第二个断点在 $b_2 \in [nxt_i+1, lst_{a_i}]$ 的位置的答案加 1。

特判 nxt 不存在的情况。

用线段树简易维护答案最大值以及方案。时间复杂度 $O(n\log n)$。

[G] Starlight

先假定将数组排序，容易归纳说明一次询问的答案为所有排名 $\le z$ 并且和 z 模 m 同余的所有数之和。对于加入和删除数的操作我们用块状数组维护每个块中排名 $\bmod m$ 为某个值的所有数之和可以做到时空 $O(n^{1.5})$。

为了做到空间线性，我们对 $m < \sqrt{n}$ 的数据这样做，而对 $m \ge \sqrt{n}$ 的数据，对于每次询问暴跳所有要加和的位置，只要做到 $O(1)$ 找到一个排名的值即可，而这是容易维护的。

[H] Rev Equation (NOI-tAUqe ver.)

考虑对于一个串形如 aOb=，紫眼会考虑什么字符串。分类讨论 a, b 的关系：

- $a = b$：唯一的会被考虑的字符串是 =aOa=。因此不合法。
- $a \neq b$：会被考虑的字符串有很多，手玩可以发现只有 a=bOb=a，aOb=bOa 这两种可能是有意义的。考虑第一种，只有形如 0-x=（$x \neq 0$）的情况才合法；第二种当 O 不是减号的时候都合法。

总结一下，当且仅当 $a \neq b$ 且（$a = 0$ 或 O 不是 -）的情况是可以的，否则不行。

[I] Matrix

先判断什么时候无解。注意到当某数 i 是 n 的倍数的时候，这个数无法在竖直方向上行走。而当它是 m 的倍数的时候，这个数无法在水平方向上行走。所以 $i = \mathrm{lcm}(m, n)$ 只能等于 nm。所以 n, m 不互质的时候无解。

接下来，构造所有 n, m 互质的情况。先把 $1 \sim m$ 放到第一行上。直接从 1 开始左右左右摇摆填充即可。

接下来，一行填满之后我们需要换到另一行。考虑和水平时一样，每次上下摇摆。此时走到的行号相对于第一行的位移是 $\{0, m, -m, 2m, -2m, ...\}$，是一个模 n 完系（由于 n, m 互质），因此一定不会回到之前填过的某一行。在新的行上继续摇摆即可。

[J] Segments

题外话：本题题目背景是真实的。

对于 $n \leq 3$，显然是无解的。

对于 $n = 4$，解就是一个正方形，正如样例；证明的话，发现如果要是一个环，必然是菱形；有直角的菱形，为正方形；这一点初中就学过了。

对于 $n = 5$，无解，正如样例，证明也不复杂：考虑固定两条边三个顶点 (0, 0, 0), (1, 0, 0), (1, 0, 1)，那么下一个点的 $z = 1$，到 $(1, 0, 1)$ 的距离为 1，到 $(0, 0, 0)$ 的距离为 $\sqrt{2}$，可以发现这个点的位置只有两种（还是对称的），钦定其为 $(0.5, \frac{\sqrt{3}}{2}, 1)$，然后发现最后一个点无法同时符合长度和垂直的限制。于是无解。

[J] Segments

对于 $n = 6$，显然有两种平凡的解，均取自一个正方体的框架：

$(0, 0, 0), (0, 0, 1), (0, 1, 1), (1, 1, 1), (1, 1, 0), (1, 0, 0)$

$(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 1, 0), (1, 1, 0), (1, 0, 0)$

但是，有点反直觉的是，对于 $n = 6$，其实有 ∞ 种解：

考虑选出将 6 个顶点分为两组，每组内两个顶点在环上不相邻（即，在环上这两组点交错）。容易发现，每一组点间连边，构成边长为 $\sqrt{2}$ 的等边三角形。同时，对于一个等边三角形的一条边，这两个顶点在环上的中间的那个顶点，应该在“以这条边的中点为圆心，$\frac{\sqrt{2}}{2}$ 为半径，圆所在平面垂直于该边，的圆上”。这样就保证了边长均为 1、相邻边相互垂直。经过暴力计算，可以发现有无数组解。

[J] Segments

对于 $n > 6$ 的情况，可以感性理解，也可以使用自由度说明：一共有 n 个长度相同的向量，每个有 2 个自由度；要求是一个环，即限制了 x, y, z 的和，减少 3 自由度；平移、旋转、反射使得重合的不被重复计算，减少 3 自由度；相邻向量要求垂直，减少 n 自由度；对于 $n > 6$，总的自由度是 $n - 6 > 0$，因此有 ∞ 组解。

[J] Segments

对于构造方案:

$n=4,5$ 样例已经给出;

n 为偶数可以轻松构造。注意到线段可以重合，因此你可以现在原地转若干圈，然后归到 $n=4,6$ 的情况。

$n=7$ 可以选择手算，考虑一个正方形少一条边，在这条缺失的边两端各接一根线段，要求他们另一端顶点距离 $\sqrt{2}$，然后再拼上一个两条边构成的直角，通过绕那条 $\sqrt{2}$ 的斜边旋转实现垂直。

$n=9$ 可以考虑删掉 $n=7$ 的那个正方形，缺失的边的对边，然后再拼上去一个缺一条边的正方形，使得其垂直。

[J] Segments

不过这个题还可以使用模拟退火来实现：具体的，我们认为误差越小，价值越高；误差可以被认为是，向量和的长度（越接近 0 越优），以及最后一个向量和第一个向量的夹角，越接近 $\frac{\pi}{2}$ 越优。

这个模拟退火能找到解的概率还是挺大的，对于误差在合理范围内的解，至多跑几十次就能找到。

[J] Segments

下面给出一组 $n = 7$ 的解：

(0.000000, 0.000000, 0.000000)

(1.000000, 0.000000, 0.000000)

(1.000000, 0.000000, 1.000000)

(0.000000, 0.000000, 1.000000)

(0.000000, 0.978318, 1.207107)

(0.691080, 1.128011, 0.500000)

(0.000000, 0.978318, -0.207107)

[J] Segments

对应拓展到 $n = 9$:

(0.000000, 0.000000, 0.000000)

(1.000000, 0.000000, 0.000000)

(1.000000, 1.000000, 0.000000)

(1.000000, 1.000000, 1.000000)

(1.000000, 0.000000, 1.000000)

(0.000000, 0.000000, 1.000000)

(0.000000, 0.978318, 1.207107)

(0.691080, 1.128011, 0.500000)

(0.000000, 0.978318, -0.207107)

[J] Segments

如果有选手对 $n = 6$ 存有异议，下面给出一组非平凡的 $n = 6$ 的解：

(0.000000, 0.000000, 0.000000)

(1.000000, 0.000000, 0.000000)

(1.000000, 0.000000, 1.000000)

(0.552786, 0.894427, 1.000000)

(0.914553, 1.075310, 0.085447)

(0.000000, 0.953830, -0.300346)

[K] Amazing Sets

首先将图建出来之后缩点，由于是树上加一些返祖边，我们可以得知缩点之后的图是一个外向树。那么我们实际是要求，在这棵树上，选出若干不交子树，总权值一共有多少种。

设计 dp 状态为 $f_{i,j} = 0/1$，表示 i 号点的子树内，选择了总权值为 j 的若干不交子树，是否可行。转移直接树上背包合并即可。

时间复杂度瓶颈为小常数 $O((n + \sum a_i) \sum a_i)$。可以通过。可以进一步优化达到更优复杂度，但这不在本题的考察范围内。

[L] Desertrium

以下均为缩点后定 1 号环为根的有根树。注意缩点前节点（称作节点）和环缩成的点（称作环）之间作区分。

首先考虑子水母。枚举每个环，对于每条出边 u 记录 f_u 代表从 u 出发不经过当前环边和父边的链的数量。这个值是好统计的（在实现中对边统计是困难的，所以我们不妨对环上的每个点记录其连接的所有出边的 f 之积）。不过环上连到父节点的边延伸出去的链为统计，所以再记录 up_x 表示环 x 经过父边延伸出去的链的数量，此时把所有边的 f 相乘即可。

[L] Desertrium

然后是子海星和子鳗鱼。中心节点度数至少为 3 时是好做的，枚举中心节点之后可以通过刚才计算得到的 f 算出结果。对于一条链来说，由于其恰好会被计算两次，我们钦定其两个端点是中心节点。换言之，我们认为中心节点的度数是非 2 的正整数。由于我们刚才计算出了 f 值，可以直接算出每个点作为中心节点的答案。需要注意的是延伸出去的链并非一定是往下的，也可能经过环边，于是我们可以维护 sum_u 表示每个环的所有出边的 f 之和。这部分需要注意贡献不要算重，剩下的都是一些 dirty work。

THANKS!

AC.NOWCODER.COM