牛客竞赛 AC.NOWCODER.COM | NIO

2022 牛客
暑期多校训练营

CVBB 命题组

难度预测

- check-in: G
- easy: A D
- medium-easy: C I
- medium: J
- medium-hard: H
- hard: B E F
- very-hard: K

统计

- A: (12) 1334 / 3358
- B: (45) 23 / 239
- C: (70) 17 / 85
- D: (30) 960 / 6414
- E: (45) 28 / 222
- F: (46) 18 / 135

统计

- G: (3) 1482 / 4184
- H: (13) 39 / 291
- I: (18) 310 / 590
- J: (16) 31 / 783
- K: (-) 0 / 36

Villages: Landlines

题目大意

- 数轴上有恰好一个发电站与 $n-1$ 个建筑物
- 在数轴上放置一些电力塔使得所有建筑物通过电力塔与发电站连通
- 能源站位于 x_s，能与距离 r_s 内的电力塔直接连通
- 第 i 个建筑物位于 x_i，能与距离 r_i 内的电力塔直接连通
- 可以消耗 $|x_A - x_B|$ 长度的电线连通位于 x_A 与 x_B 的电力塔
- 最小化消耗的电线长度
- $1 \le n \le 2\times 10^5$, $-10^9 \le x_s, x_i \le 10^9$, $1 \le r_s, r_i \le 10^9$

题解

- 等价于有 n 个区间 $[x_s - r_s, x_s + r_s], [x_i - r_i, x_i + r_i]$
- 定义两个区间连通当且仅当两个区间交集非空
- 可以任意添加区间
- 求使得 n 个区间连通需要添加的最小区间长度和

题解

- 先合并所有给出区间中连通的区间
- 可以得到若干个大区间
- 填补大区间之间的空隙即可
- $\mathcal{O}(n \log n)$

Spirit Circle Observation

题目大意

- 给定长度为 n 的数字串 s，计算满足下列条件的二元组 (A,B) 个数：
  - A,B 是 s 的子串；
  - A,B 长度相同；
  - $B=A+1$。
- 即使有两个子串相同，只要它们在原串位置不同，则认为是不同的子串。
- $1 \le n \le 4 \cdot 10^5$。

题解

- 可以发现满足条件的子串一定是 $A=\overline{xp99\cdots9}$，$B=\overline{x(p+1)00\cdots0}$，其中前缀 x 是一个相同的数字串（可以为空），$p\in\{0,1,\ldots,8\}$，且后面的 $99\cdots9$ 与 $00\cdots0$ 长度相同（可以为空）。因此枚举 p 的值后，实际只需要考虑对固定长度的 $99\cdots9$ 和 $00\cdots0$，计算所有可能的前缀 x 的贡献。

题解

- 可以用 SAM 求出 fail 树，然后枚举所有可能的 p。我们考虑枚举 $99\cdots9$ 和 $00\cdots0$ 的长度，并将形如 $\overline{p_1 99\cdots9}$ $(p_1 \neq 9)$ 和 $\overline{p_2 00\cdots0}$ $(p_2 = p_1 + 1)$ 的 p_1, p_2 的位置视为关键点。可以发现对所有长度而言，关键点的总数是 $\mathcal{O}(n)$ 的。对于一组相同的长度，可以在 fail 树的某个节点统计其底下 p_1, p_2 类关键点的贡献，这两类关键点在该节点能取到的所有前缀 x 正是该节点的 SAM 子串等价类，维护该信息是 $\mathcal{O}(1)$ 的。
- 对每种长度，建立虚树跑上述信息即可。
- 综上，总时间复杂度是 $\mathcal{O}(n\Sigma + n\log n)$ 的，其中 $\Sigma = 10$ 表示字符集大小。

Grab the Seat!

题目大意

- 二维平面，屏幕是 $(0,1)$–$(0,m)$ 的线段
- 有 n 行 m 列座位在屏幕前面，是坐标范围 $1 \le x \le n, 1 \le y \le m$ 的整点
- 有 k 个座位已经有人，求出到屏幕的视线不被任何人挡住的座位数量
- q 次询问，每次修改一个人的坐标后求出答案
- $2 \le n, m \le 2 \times 10^5$, $1 \le k \le 2 \times 10^5$, $1 \le q \le 200$

题解

对于 $y=1$ 或 $y=m$ 的列，每个人只会挡住自己正右方的人。特殊处理一下。

而对于 $y=2,3,\ldots,m-1$，每个人挡住的区域是 $(0,1)$ 和 $(0,m)$ 到这个点的两条射线之间的区域。两条射线的方向都是 x 坐标增大，y 坐标一个增大一个减小。最后不合法的区域就是这些区域的并集。

整个区域实际上构成一个折线图。对于每个 y，会有一个分界线 t，在这一列上，$x \in [1,t)$ 的点是满足条件的，而 $x \in [t,n]$ 的点会被挡住。

题解

考虑如何维护出这个折线图。

本来 std 有 set 的做法，但是验题的时候被一眼李超树了，所以加了询问卡掉了带 $\log$ 的做法。要求每次询问线性完成。

题解

折线图由 $2k$ 个线段构成。这 $2k$ 条线段有比较明显的性质，就是延长线都经过 $(0,1)$ 或 $(0,m)$。可以把它们分成两类。

那么实际上可以对于延长线经过这两个点的线段分别维护一下斜率。

首先对每个 y，如果这一列有多个人，那么肯定考虑 x 最小的人，因为小的会完全覆盖大的。

然后对于 $(0,1)$ 和 $(0,m)$ 分开考虑。

题解

以 $(0,1)$ 为例，所有线段都是从某个点往右上方走。

如果按 y 从小到大加入线段，不难发现斜率大的线段一旦加入，就一直会覆盖斜率小的线段。对于某个 y，要找到最小的合法 x，那么肯定取的是前面所有线段中斜率最大的线段。所以对 y 正着扫一遍，在扫的同时维护一下最大的斜率，即可得出每一列只考虑经过 $(0,1)$ 的线段的答案。

$(0,m)$ 同理，倒着扫一遍维护斜率最小（绝对值最大）的线段即可。每个 y 在两种情况取 $\min$ 就是答案。

总复杂度为 $\mathcal{O}(mq)$。

Mocha and Railgun

题目大意

- 给定一个圆和严格位于圆内的一点 P
- Mocha 会从点 P 向任意角度发射一个长度为 $2d$ 的电磁炮
- 电磁炮底边的中点为点 P 且两端位于圆内
- 询问单次发射能摧毁的最大圆弧长
- $1 \leq T \leq 1000, -10^9 \leq x, y \leq 10^9, 1 \leq r, d \leq 10^9$

题解

无论当前的电磁炮旋转角度如何，我们可以固定电磁炮的方向，将点 Q 绕原点旋转，从而使得电磁炮方向竖直向上（即 y 轴正方向）。

那么我们可以将题目转化成为电磁炮的方向总是竖直向上的，点 Q 绕原点旋转一周的过程中可以摧毁的最长墙壁长度。并且题目保证了 base segment 总是位于圆内部的，那么我们设点 Q 到原点距离是 dis，点 Q 绕原点旋转一周就可以转化成点 Q 在以 $(-dis, 0)$ 和 $(dis, 0)$ 为端点的线段上移动。

题解

　　显然摧毁的墙壁是一段劣弧，那么长度的大小关系就可以映射到弦长的大小关系。设电磁炮摧毁墙壁的端点为 P_L 和 P_R，则对应弦长长度为 $\sqrt{(y_{P_L}-y_{P_R})^2+(2d)^2}$，其中 d 是给定的，那么显然两点纵坐标之差的绝对值最大时弦长最长，对交点坐标求导不难看出当 Q 位于线段端点时差的绝对值最大，返回原题面，也就是 base segment 位于直线 QO 上时摧毁墙壁最长。

LTCS

题目大意

- 定义一棵带点权的有根树 T_a 是 T_b 的子序列当且仅当满足以下条件:
  - T_a 的点到 T_b 的点存在一个单射 F
  - 对于 T_a 中的所有点 u 有 $c_{a,u} = c_{b,F(u)}$
  - 如果 u 在 T_a 中是 v 的祖先，则 $F(u)$ 在 T_b 中是 $F(v)$ 的祖先
  - 如果 u 在 T_a 中是 v, w 的父亲，则在 T_b 中 $F(u)$ 在 $F(v), F(w)$ 的简单路径上
- 求两棵以 1 为根的树 T_1, T_2 的最大公共子序列大小
- $2 \le n, m \le 500$

题解

设 $dp_{u,v,0/1}$ 表示第一棵树以 u 为根的子树，第二棵树以 v 为根的子树，u 和 v 不一定匹配/匹配的答案。

对于 $dp_{u,v,1}$，考虑 T_3 中 u,v 对应结点的所有儿子，每个儿子都应该对应 T_1 中 u 的不同儿子的子树，T_2 同理，否则将违反条件 3。那么将 u 的所有儿子 u_1 和 v 的所有儿子 v_1 建立一个二分图，边权为 $dp_{u_1,v_1,0}$，答案显然为最大权匹配 $+1$。

对于 $dp_{u,v,0} = \max_{u_1 \text{ is child of or is } u, v_1 \text{ is child of or is } v} dp_{u_1,v_1,1}$。最终答案是 $dp_{1,1,0}$。

题解

$dp_{u,v,1}$ 的复杂度看似是 $\mathcal{O}(n^5)$，因为需要枚举两两点，并进行 $\mathcal{O}(n^3)$ 的匹配，但是复杂度实际是 $\mathcal{O}(n^3)$。

此题做匹配时两侧点数不同，设为 $|U|,|V|$，我们开始仅需要枚举 $\min(|U|,|V|)$ 个点进行增广，因此匹配的总复杂度是不超过 $\min(|U|,|V|)(|U|+|V|)^2$。

题解

不妨将 T_1, T_2 的所有结点的儿子数量拿出来排序后变成一个数列，设为 a，数列长度是 $N = n + m$。

假设两两要做一次匹配，总复杂度为

$$\mathcal{O}\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} a_i \times (a_i + a_j)^2\right) = \mathcal{O}\left(\sum_{i=1}^{N}\sum_{j=i+1}^{N} a_i \times (a_i^2 + a_j^2)\right)$$

$$\leq \mathcal{O}\left(\sum_{i=1}^{N} a_i \times \sum_{j=i+1}^{N} a_j^2\right) = \mathcal{O}(N \times N^2) = \mathcal{O}(N^3)$$

这显然比真实复杂度高，因此总复杂度是 $\mathcal{O}((n+m)^3)$。$dp_{u,v,0}$ 的复杂度看似是 $\mathcal{O}(n^4)$，但是实际上也是 $\mathcal{O}(n^2)$ 的，证明方法类似。

Cut

题目大意

- 给定 1 到 n 的排列 $a_1, a_2, \ldots, a_n$，有 m 次以下三种操作：
  - 将 $a_l, a_{l+1}, \ldots, a_r$ 从小到大排序。
  - 将 $a_l, a_{l+1}, \ldots, a_r$ 从大到小排序。
  - 询问 $a_l, a_{l+1}, \ldots, a_r$ 中最长的奇偶交替子序列。
- $1 \leq n, m \leq 10^5$。

题解

- 先考虑一个弱化问题，将第三种操作改为询问区间和。
- 当一个区间被排序后，我们将它们合并在一起，称为连续段，初始时每个位置单独属于一个连续段。
- 对于一个连续段，使用一个动态开点权值线段树存储其中的所有值，具体来说，如果该连续段中存在某个值，则线段树中存在对应值的叶子节点。因为连续段中的数字是有序的（正序或逆序），因此该线段树同时保存了连续段中数字的顺序，从而连续段中某个区间可以对应到该线段树上某个区间。

题解

- 对于前两种操作，将所有完整位于操作区间的连续段进行线段树合并；对于不完整部分位于操作区间的连续段，使用线段树分裂将需要的区间分裂出来再进行线段树合并。
- 对于每个连续段，将其区间和（也就是对应线段树中所有叶子节点对应值的和）放到其左端点的位置上，再额外开一个非动态开点的普通线段树 T 维护这些位置上的区间和。
- 对于第三种操作，对于所有完整位于操作区间的连续段，可以使用 T 进行区间查询；对于不完整部分位于操作区间的连续段，可以在对应连续段对应线段树上查询所需区间得到结果。

题解

- 最后，回到原始问题，设 $f_{i,j}$ 表示一个区间首尾的奇偶性分别为 $i=0/1, j=0/1$ 时，最长的子序列长度。将上述过程中所有的区间和均替换为 $f_{i,j}$，即可解决原始问题。
- 线段树合并的运行次数不超过总节点数，初始共有 $\mathcal{O}(n\log n)$ 个节点，线段树分裂产生的节点不超过 $\mathcal{O}(m\log n)$ 个，因此时空复杂度均为 $\mathcal{O}\big((n+m)\log n\big)$。

Lexicographical Maximum

题目大意

- 给定 n，将 $1, 2, \ldots, n$ 视为不含前导零的字符串
- 求这些字符串中字典序最大的字符串
- $1 \le n \le 10^{1000000}$

题解

- 显而易见，答案除去最后一位，其余位均为 9
- 若 n 除去最后一位，其余为均为 9 则答案即为 n
- 若不然则答案为 $|n|-1$ 个 9
- 其中 $|n|$ 指 n 对应字符串的长度
- $\mathcal{O}(|n|)$

Fly

题目大意

- n 种物品每种有无数个，第 i 个物品的体积为 a_i，求解选择物品总体积不超过 M 的方案数。
- 此外，有 k 个限制，第 i 个限制形如第 b_i 个物品所选的数量二进制表示下从低到高第 c_i 位必须为 0。
- $1 \le n \le 4 \times 10^4$，$0 \le M \le 10^{18}$，$1 \le k \le 5 \times 10^3$，$1 \le a_i \le 4 \times 10^4$，$\sum_{i=1}^{n} a_i \le 4 \times 10^4$。

题解

- 我们可以认为 n 和 $\sum_{i=1}^{n} a_i$ 是同阶的，并令 m 是最小的非负整数满足 $2^m > M$，显然 $m = O(\log M)$。此外，设 k 个限制对应的二元组 (b, c) 组成的集合为 S。
- 这 k 个限制都是针对二进制某一位的，因此我们可以从这里切入，使用类似二进制分组的方法将完全背包转化为连续做 m 个 01 背包，其中第 i 个 01 背包由 n 个物品组成，其中第 j 个物品的体积为 $2^i \times a_j$，这里 $i = 0, 1, \ldots, m-1$。

题解

- 但这样不满足 k 个限制，我们再额外对其进行改动，对于第 i 个 01 背包，只考虑 $(i,j) \notin S$ 的 j，这样连续做改动后的 m 个 01 背包所得到的解就和原问题等价了。
- 当进行到第 i 个 01 背包的时候，设当前所选物品体积为 V，对于低位的情况，我们只关心 $V \bmod 2^i$ 和 $M \bmod 2^i$ 的大小关系，因此只需要记录 $\left\lfloor \frac{V}{2^i} \right\rfloor$ 和 $\left[(V \bmod 2^i) > (M \bmod 2^i) \right]$ 的值。

题解

- 设 $f(i,j,b)$ 为第 i 个 01 背包进行完毕，选择物品总体积为 V 的方案数，其中 V 满足 $j=\left\lfloor\frac{V}{2^i}\right\rfloor$ 且 $b=\left[(V \bmod 2^i)>(M \bmod 2^i)\right]$。那么最终答案即为 $f(m,0,0)$。
- 关于 j 的范围，可以发现其最值为 $\sum_{l=0}^{i}\frac{n}{2^l}<2n$。

题解

- 设 $g(i,j)$ 为第 i 个 01 背包选择物品总体积为 $j\times 2^i$ 的方案数。那么有:

$$g(i,j)=[x^j]\prod_{(l,i)\notin S}(1+x^{a_l})=[x^j]\frac{\prod_{l=1}^{n}(1+x^{a_l})}{\prod_{(l,i)\in S}(1+x^{a_l})}$$

题解

- 我们可以先用分治 NTT 在 $O\left(n\log^2 n\right)$ 的时间内求出多项式 $h(x)=\prod_{l=1}^{n}(1+x^{a_l})$，之后对于每个 i，用其除以 $\prod_{(l,i)\in S}(1+x^{a_l})$ 得到对应的 $g(i,j)$，这可以通过 $O(k)$ 次逆向 01 背包来实现，时间复杂度为 $O(nk)$。

题解

- 考虑 b 的变化，相当于两个二进制串 s 和 t 进行比较，最高位分别为 x 和 y，除去最高位的大小关系是 z，那么 $[s>t]=[x>y\lor(x=y\land z)]$，设 $B(x,y,z)=[x>y\lor(x=y\land z)]$。
- 最后，我们有如下转移式，其中 M_i 是 M 在二进制表示下从低到高第 i 位的值：

$$f(i,j,b)\times g(i,l)\to f\left(i+1,\left\lfloor\frac{j+l}{2}\right\rfloor,B((j+l)\bmod 2,M_i,b)\right)$$

这可以通过 $m+1$ 次卷积完成，因此总时间复杂度为 $\mathcal{O}\left(n\log^2 n+nk+n\log n\log M\right)$。

Chiitoitsu

题目大意

- 初始手牌有 13 张麻将牌，相同牌至多出现 2 张
- 每轮可以从牌堆摸牌，若达成七对子则自摸胡牌
- 若不然则选择手牌中某张牌并丢弃之
- 给定初始手牌，求最优策略下达成七对子的期望轮数
- 多组数据，数据组数不超过 10^5 组

题解

由于初始手牌中每种牌最多两张，因此最优策略是，若摸到的牌能凑成对子则丢弃单牌，否则丢弃摸到的牌。

考虑 DP 求期望，令 $f_{s,r}$ 表示当前手牌中有 s 张单牌且牌堆中剩余 r 张牌时达成七对子的期望轮数，则有：

$$f_{s,r} = \begin{cases} 1 + \frac{r-3}{r} f_{1,r-1} & (s = 1) \\ 1 + \frac{3s}{r} f_{s-2,r-1} + \frac{r-3s}{r} f_{s,r-1} & (s > 1) \end{cases}$$

对于给定的初始手牌，设其单牌数量为 s_0，则 $f_{s_0,136-13}$ 即为答案。

时间复杂度 $\mathcal{O}(7 \times 136 + T)$。

Serval and Essay

题目大意

- 有一张 n 个点 m 条边的无重边无自环的有向图
- 初始时可以选择一个点染黑，其余点均为白点
- 若某个点所有入边的起点均为黑点，则该点可以被染黑
- 最大化图中黑点数量
- 多组数据，$1 \le \sum n \le 2 \times 10^5, 1 \le \sum m \le 5 \times 10^5$

树关系

记 S_u 为 u 作为初始黑点时迭代得到的最终黑点集合，并记 I_u 为 u 所有入边的起点构成的集合。

显然地，若 $v \in S_u$ 则有 $S_v \subseteq S_u$。更进一步，有以下定理：

Theorem

若 $S_u \cap S_v \neq \emptyset$ 则有 $S_u \subseteq S_v$ 或 $S_v \subseteq S_u$。

我们来证明这一点。

树关系

Proof.

假设 $S_u \cap S_v \neq \emptyset$。$u = v, v \in S_u, u \in S_v$ 是朴素情况，定理显然成立。

不失一般性地，考虑 S_u，存在推演序列 $w_1 = u, w_2, \ldots, w_{|S_u|}$ 使得对于除 w_1 外任意 w_i 满足 $I_{w_i} \subseteq \{w_j \mid j < i\}$，于是存在最小的 i 使得 $w_i \in S_u \cap S_v$，记这个 i 为 k。

仅需考虑 $w_k \neq u$ 且 $w_k \neq v$ 的情况。由于 $w_k \neq v$，因此 $w_k \in S_v$ 当且仅当 $I_{w_k} \in S_v$，而任取 $w \in I_{w_k}$ 均有 $w \in \{w_j \mid j < k\} \neq \emptyset$，而这与 k 为满足条件的最小值矛盾，即知定理成立。 $\square$

树关系

将 $S_1, \ldots, S_n$ 视作结点，若 $S_u \subseteq S_v$ 则要求 S_v 成为 S_u 祖先，可以得到唯一的森林。

我们只需要求出森林中每棵树的大小，最大值即为答案。

启发式合并

一种可行的解法是加速迭代过程，迭代过程描述如下。

初始时对于所有点 u 置 $S_u = \{u\}$，并置 $T = V$。

每次考虑所有 T 中的点 v，若存在 T 中的点 u 使得 $I_v \subseteq S_u$，则令 $S_u \leftarrow S_u \cap S_v$，并令 $T \leftarrow T \setminus \{v\}$，重复到不存在满足条件的 u, v 为止。

显然地，任意时刻下对于 T 中的任意 u, v 而言 $S_u \cap S_v = \emptyset$。

启发式合并

考虑如何实现这一过程。我们将所有 S_u 看作一个点，S_u 向 S_u 中所有点的出边终点除去自身的所属集合连边。当 $I_v \subseteq S_u$ 满足时，将 S_u, S_v 按集合大小启发式合并为同一个代表 S_u 的点，同时合并其出边。容易发现，$I_v \subseteq S_u$ 当且仅当在合并后 S_v 仅有一条来自 S_u 的入边。最终 T 中的点 u 对应的 S_u 即对应森林中一棵树的根结点，取大小最大值即可。

原图中每个点最多被合并 $\log n$ 次，利用 set 维护即可在 $\mathcal{O}((n+m)\log^2 n)$ 的时间内求出最终森林。

随机化

另一种可行的解法是随机化。

我们考虑以随机排列 p 遍历所有结点，并记第 i 个遍历的结点为 p_i。当且仅当 $p_i \notin S_{p_1} \cap \cdots \cap S_{p_{i-1}}$ 时，从 p_i 出发遍历原图求出 S_{p_i}。显而易见，森林中每棵树的根结点对应的 S_u 必定会被计算，因此可以求出答案。

事实上，此解法的期望时间复杂度是 $\mathcal{O}(m\log n)$，我们将证明这一点。

随机化

考虑点 u，其出边在计算 S_u 的某个祖先 S_v 时会被遍历。我们尝试估计 u 出边被遍历的次数。

不妨设深度为 x 的 S_u 出边被遍历的期望次数为 $f(x)$，容易写出：

$$\begin{cases} f(x) = 1 + \dfrac{1}{x}\displaystyle\sum_{i=1}^{x-1} f(i) \\ f(1) = 1 \end{cases}$$

随机化

粗略地估计，不妨考虑以下方程及初值条件：

$$\begin{cases} g(x) = 1 + \dfrac{1}{x}\displaystyle\int_1^x g(x)\,\mathrm{d}x \\ g(1) = 1 \end{cases}$$

容易解出 $g(x) = 1 + \ln x$，且 $g(x) > f(x)$ 对于 $x > 1$ 成立，即知深度为 x 的 S_u 出边被遍历期望 $\mathcal{O}(\log x)$ 次，因此期望时间复杂度是 $\mathcal{O}(m \log n)$ 的。

Villages: Landcircles

题目大意

- 给定二维平面 n 个圆
- 可以在圆外任意处添加额外的点
- 连接圆和点，圆和圆，点和点所需的花费为连接所用线段的长度
- 询问将 n 个圆连通的最小费用
- $2 \leq n \leq 9, -2000 \leq x, y \leq 2000, 1 \leq r \leq 2000$

相关知识

- 费马点
- Gilbert E N, Pollak H O. Steiner minimal trees[J]. SIAM Journal on Applied Mathematics, 1968, 16(1): 1-29.
- Hwang F K. A linear time algorithm for full Steiner trees[J]. Operations Research Letters, 1986, 4(5): 235-237.

平面欧几里得斯坦纳树

显然所有点应当连成一棵树形结构。不妨假设所有圆的半径趋于无穷小，那么这就是经典的平面欧几里得斯坦纳树问题。可以参看 [Gilbert] 了解一些基本的概念。先介绍一些名词：

- 初始点：输入中原有的点
- Steiner 点：Steiner 树中额外添加的点
- Steiner 树：通过添加 Steiner 点将所有初始点连通所形成的树形结构
- Full Steiner 树：Steiner 树的一个部分结构，每个初始点的度为 1，详见下文

平面欧几里得斯坦纳树

Lemma (1)

Steiner 点的度必须为 3，且任意两个邻点 A, B 与 *Steiner* 点 S 所成的角 $\angle ASB = 120°$。

Proof of Lemma 1

假设 $\angle ASB < 120°$。分两种情况讨论，若 $\angle SAB$ 和 $\angle SBA$ 均小于 $120°$，那么 $\triangle SAB$ 中将存在一个费马点 P，使得 $PS + PA + PB < SA + SB$，那么引入新的 Steiner 点 P，将其向 S, A, B 连边将使得答案更优。

类似地，不妨设 $\angle SAB \geq 120°$，那么 $SA + AB < SA + SB$，将 SB 删去替换成 AB 将使得答案更优。因此，$\angle ASB \geq 120°$。这使得 Steiner 点的度数至多为 3。

Steiner 点度为 1 时显然没有意义。Steiner 点度为 2 时，只有其在两个相邻点连线上时才能达到最优，但此时不加这个点也能得到同样的答案。因此，Steiner 点的度必须为 3，而 $\angle ASB$ 必须恰为 $120°$。

平面欧几里得斯坦纳树

读者可能会发现，初始点也有类似的性质。但是这一性质在原问题中并不一定成立。

Lemma (2)

对于任意一个 n 个点的点集，其 *Full Steiner* 树的点数为 $2n-2$，不同的 *Full Steiner* 树有 $\frac{(2n-4)!}{2^{n-2}(n-2)!}$ 种。

平面欧几里得斯坦纳树

Proof.

注意到 Steiner 点和初始点的度数分别为 3 和 1，而树所有点的总度数为 2 倍点数减 2，解方程即可知道有 $n-2$ 个 Steiner 点。
考虑这棵树的 Prüfer 序列，其共有 $2n-4$ 项，且仅包含所有 Steiner 点的标号，每个标号出现 2 次，注意到不同的 Steiner 点是不可区分的，因此共有 $\frac{(2n-4)!}{2^{n-2}(n-2)!}$ 种不同的 Full Steiner 树。 □

注意到在本题的数据范围（$n=9$）下，枚举所有 Full Steiner 树是可以接受的，约 1.35×10^5 个。

算法

考虑任意一棵 Steiner 树，其中可能存在度超过 1 的初始点。不妨将这样的初始点拆成度数份，每份分别拥有一条原来的边。容易发现，这样每个连通块都是一棵 Full Steiner 树。因此，任意的 Steiner 树可表示为若干 Full Steiner 树的集合。换言之，每个子集可求出对应该子集的 Full Steiner 树，Steiner 树可表示为若干子集的 Full Steiner 树的并，这些子集必须要是连通的（即任意两个子集如果有交连边，所得图必须连通）。显然每个子集内部，需求出最小的 Full Steiner 树。

算法

这很自然地引出了一个状态压缩 dp 的做法。记 dp_S 表示将点集 S 连通的最小 Steiner 树，那么有转移方程 $dp_{S\cup T} \xleftarrow{\min} dp_S + F_T$，其中 S 和 T 必须恰有一个公共点，F_T 表示 T 的最小 Full Steiner 树。可以证明，这一部分的时间复杂度可以做到 $\mathcal{O}(3^n n)$。当然，这部分并非主要复杂度，差一些的算法也可以通过。

算法

剩余的主要问题就是求出某个点集对应的最小 Steiner 树。要枚举所有 Full Steiner 树的拓扑，可以首先枚举**引理 2** 中提到的 Prüfer 序列，这很容易在线性时间完成。接下来要将每个 Prüfer 序列在线性时间内重建出原本的树。对于这一棵树，需要在线性时间内求出其对应的最小 Full Steiner 树。即，目前各个点之间的连接方式已经确定，但是还需要再确定新增的 Steiner 点的坐标，以求得最终答案。这一算法请参见 [Hwang]，文章不长，这里就不赘述了。

复杂度分析

时间复杂度为 $3^n \cdot \text{poly(n)} + \sum_{i=3}^{n} i\binom{n}{i}\frac{(2i-4)!}{2^{i-2}(i-2)!}$。

考虑和式内相邻两项 i 和 $i+1$ 的比值为 $\frac{i}{(2i-3)(n-i)}$，当 n 充分大，且 $i < n-1$ 时，这一比值是小于 $1/2$ 的。根据等比数列求和公式，可知总复杂度与 $i = n$ 时的复杂度同阶。

复杂度分析

根据 Stirling 公式，有：

$$
\begin{aligned}
n \cdot \frac{(2n-4)!}{2^{n-2}(n-2)!} &\sim n \cdot \frac{\sqrt{2\pi(2n-4)}((2n-4)/e)^{2n-4}}{2^{n-2}\sqrt{2\pi(n-2)}((n-2)/e)^{n-2}} \\
&\sim n2^{n-2}((n-2)/e)^{n-2} \\
&\sim (2n/e)^{n-1}
\end{aligned}
$$

显然加号左边这一项不是主要复杂度。因此，该算法的时间复杂度是 $\mathcal{O}((2n/e)^{n-1})$。

圆的斯坦纳树

现在回到原问题。事实上，圆的 Steiner 问题和点的 Steiner 问题非常接近，我们将通过下面的引理证明这一点。

Lemma (3)

原问题的最优方案能够与圆心（作为点时）的一棵 *Steiner* 树对应，且所有 *Steiner* 点不在圆内。

Proof of Lemma 3

若某两个圆之间有连边，注意到这条边一定是两个圆的圆心间连线删去圆内的部分，那么对于这样的边，可以把其延长到圆心。这样显然所有的点仍然是连通的。这样就得到了一棵 Steiner 树。

Proof of Lemma 3

对于圆外的 Steiner 点，**引理 2** 仍然成立。注意我们这里不考虑从圆心连出来，只考虑圆外的部分。可以理解成点 A, B 是 Steiner 树在圆周上的交点。考虑在证明过程中添加的新 Steiner 点或新线段，它们确实可能在圆内，从而违反了题目要求，但是显然删去圆内的部分只会使答案更小而连通性不变。因此最优方案下所有的 Steiner 点度为 3，与邻点成 120° 角，且一定不会在圆内。这一性质对初始点不成立的原因是它们在圆内，因此无法套用删去圆内部分的分析方法。事实上，考虑一个极大的圆周围有多个极小的圆，显然由大圆向每个小圆连边是最优的。

圆的斯坦纳树

我们还会发现一些有趣的性质。对圆心的任意一棵 Steiner 树，其在原问题上的代价即为所有边的长度之和减去圆内部线段的长度之和。当所有 Steiner 点均在圆外时，需要减去的代价恰是每个初始点的度数乘上对应圆半径之和。而这个代价正好可以分拆到每个子集，即每个子集的答案是 Full Steiner 树的长度和减去每个元素的半径之和。因此，我们几乎不用修改原做法，仅需把半径减去即可。

圆的斯坦纳树

另外，在这种做法下，甚至可以不用检查 Steiner 点是否在圆内。即使枚举到的 Steiner 点在圆内，也只会使得答案更差。假如这个 Steiner 点没有跟初始点相连，那么它在圆内的部分没有被减掉，那么这样算出的答案要比真实答案大。假如它跟某个初始点相连，但是未在任何一个与其相邻的初始点的圆内，情况也是相同的。而如果其在某个相邻的初始点的圆内，可以发现要使图连通，这个 Steiner 点所对应在圆内的部分长度一定大于等于半径，而我们只减掉了一个半径，所以答案也变差了。

综上所述，圆的 Steiner 问题和点的 Steiner 问题几乎相同，只需在计算 Full Steiner 树的答案时减去点集内部的半径即可。

牛客竞赛
AC.NOWCODER.COM

NIO

THANKS!

AC.NOWCODER.COM