世界是个动物园

简化题意

给出一个大小为n的竞赛图，其中点i与点$1..i-1$之间的边的方向由给出的s_i个区间确定，对于$x \in [1, i-1]$，如果存在$j(1 \le j \le s_i)$满足$x \in [l_{i,j}, r_{i,j}]$，那么i与x之间的方向就是从i到x，否则就是从x到i。对于$i = 1..n$，问保留点$1..i$后，若将任意一个三元环内的节点都并到同一个集合内，图中共有多少个集合。

$n \le 2 \times 10^5, \sum_{i=1}^{n} s_i \le 2 \times 10^6$

解法

初步分析

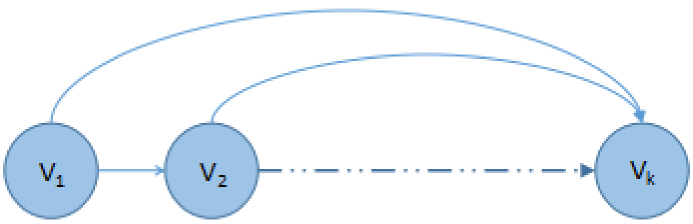
题目中给出的图是一个[竞赛图(Tournament)]，在竞赛图相关的问题上，有许多有趣的性质是值得考量的。

现在有一个竞赛图$G = (V, E)$.

性质：G中的所有强连通分量(Strongly Connected Component)缩点后的图的拓扑序是唯一的。

记k为G中的强连通分量数量，$S_1, S_2, \cdots S_k$为这k个强连通分量的拓扑序，其中$S_i = (V_i, E_i)$。

如图：



对于$i, j(1 \le i < j \le k)$，$\forall x \in S_i, y \in S_j$，都有$(x, y) \in E$.

对于一个竞赛图$G = (V, E)$，若任意一个三元环内的节点都并到同一个集合，那么集合的数量实际上就是图中强连通分量的数量。

考虑G中的k个强连通分量$S_1, S_2, \cdots, S_k$，首先，不同的强连通分量的点不会在同一个集合内，因为不存在一个三元环中的点是在不同的强连通分量内的。

然后只需要证明，一个强连通分量内的所有点最后会在同一个集合内。

证明：一个强连通竞赛图$S = (V, E)$内的所有点属于同一个集合

考虑归纳证明。

当$|V|=1$时，只有一个点，显然成立。

当$|V|=2$时，由于不存在大小为2的强连通竞赛图，同样成立。

当$|V|=3$时，有且仅有一个三元环，显然成立。

考虑$|V|>3$的情况，假设上面结论对于$|V_0|<|V|$的强连通竞赛图$S_0=(V_0,E_0)$都成立。

任取V中的一个点x，记$V'=V\setminus\{x\}$，$G'=S(V')$为S中V'的导出子图，k为G'中的强连通分量数量，$S'_1,S'_2,\cdots,S'_k$为G'中的k个强连通分量按照拓扑序排列的序列。

如果$k=1$（即G'是一个强连通竞赛图），由于S是一个强连通分量，那么在G'中一定可以找到两个点y,z，使得$(y,x),(x,z),(z,y)\in E$，如果没有这样的两个点，那么要么k大于1（即G'不是一个强连通竞赛图），要么S不是强连通的。

如果$k>1$，那么在S'_1中一定存在一个点l使得$(x,l)\in E$，在S'_k中一定可以找到一个点r使得$(r,x)\in E$，对于$i(1<i<k)$，有两种情况：

1、如果S'_i中存在一个点y使得$(x,y)\in E$，那么由于$(y,r)\in E$，三元环(x,y,r)存在，那么x与S'_i中的点属于同一个集合。

2、如果S'_i中存在一个点y使得$(y,x)\in E$，那么由于$(l,y)\in E$，三元环(y,x,l)存在，那么x与S'_i中的点属于同一个集合。

这两种情况中显然至少会有一种存在。

然后，对于S'_1和S'_k，由于$(l,r)\in E$，三元环(x,l,r)存在，所以x与S'_1,S'_k中的点属于同一个集合。

综上，一个强连通竞赛图$S=(V,E)$内的所有点都属于同一个集合。

至此，问题变成了，**对于$i(1\le i\le n)$，定义$V_i=\{x|1\le x\le i\}$，求出G中V_i的导出子图$G(V_i)$中的强连通分量的数量。**

$n\le 500$

只需要每次用tarjan求出强连通分量数即可，时间复杂度$O(n^3)$。

$n\le 2000$

考虑加入第i个点。

记k为此时图中强连通分量的数量，$S_1,S_2,\cdots,S_k$为图中所有强连通分量按拓扑序排列的序列。

找到最大的r使得S_r中存在x使得$(x,i)\in E$，如果不存在这样的r，那么$r=0$

找到最小的l使得S_l中存在x使得$(i,x)\in E$，如果不存在这样的l，那么$l=k+1$

若$l\le r$，那么再加入点i之后，$S_{1..l-1}$和$S_{r+1..k}$不会发生变动，而$S_{l..r}$会和点i合并成一个新的强连通分量。

否则，$l>r$，那么i独自一个点称为一个新的强连通分量，并且排列在S_{l-1}和S_l之间。

每次只需要$O(n)$的时间找到l, r，总的时间复杂度$O(n^2)$。

进一步分析

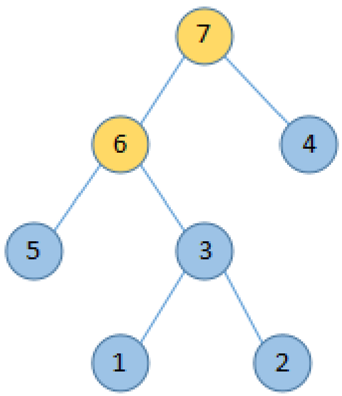
为了解决后面的部分，需要先来考虑一些特殊的性质。

实际上，从$n \le 2000$的做法里可以看出，一共会得到n个不同的强连通分量，实际上这些强连通分量之间形成了一个树形关系。

具体而言，在加入点i之后，当$l \le r$时，$S_{l..r}$与i组成了一个新的强连通分量S'，定义S'的根为i，然后树中i的儿子按照顺序从左到右就是$S_{l..r}$的根，当$l > r$时，i独自形成一个新的强连通分量S'，S'的根为i，且i是一个叶子（没有儿子）。

注意树中的一个节点的儿子是存在顺序关系的。

举个例子，如下图：



图中黄色点表示还未加入的节点，蓝色点表示已经加入图中的节点，现在要加入节点$i = 6$，此时有三个强连通分量，按照顺序，$S_1 = \{5\}, S_2 = \{3, 1, 2\}, S_3 = \{4\}$，加入节点6之后，$S_1$和$S_2$会合并成一个新的强连通分量$\{6, 5, 3, 1, 2\}$

由于产生了树状结构这一优美的性质，在暴力算法中，我们要求的l, r，l实际上就是i的出边中dfs序最小的节点所在的强连通分量，r对应的是i的入边中dfs序最大的节点所在的强连通分量。

于是，可以得到一个很自然的思路：维护$1..i$中所有节点的dfs序的相对顺序。

$n \le 10^5, S \le 3 \times 10^5$

用一棵线段树维护区间内dfs序最小的节点和dfs序最大的节点，用一棵平衡树维护已加入节点的dfs序，用一个链表维护当前的强连通分量按拓扑序排列的序列。

加入节点i时，枚举出边中的s_i段区间，询问这些区间中dfs序最小的节点，如此可以得到l，类似地可以得到r。

如果$l > k$，那么将i插入到平衡树的最末端，否则将i插入到平衡树中S_l的根的前面。

然后更新平衡树中包含i的区间的信息。

每次比较两个节点dfs序的大小关系需要$O(\log n)$的时间在平衡树上查询一个节点的排名。

时间复杂度是$O((n + S) \log^2 n)$。

$n \le 2\times 10^5, S \le 2\times 10^6, ty=0$

同样用平衡树维护节点的dfs序。

与上面做法的不同的地方在于，离线的时候可以在加入i点的时候维护所有右端点在i的询问：维护一个单调栈，栈中的元素x表示x是当前以x为开头的后缀中dfs序最小（大）的元素，加入i的时候，可以不断的比较并弹出栈顶元素，对于一次询问，只需要在栈中二分即可。

时间复杂度$O((n+S)\log n)$。

$n \le 2\times 10^5, S \le 2\times 10^6, ty=1$，**512MB**

在离线算法的基础上，用可持久化数据结构维护栈中元素的变化。

也可以求出i前第一个dfs序小（大）于i的节点，即加入i时的栈顶元素，容易发现这个也是树形结构，用倍增维护i的$\log n$个父亲即可。

时间复杂度$O((n+S)\log n)$，空间复杂度$O(n\log n)$.

$n \le 2\times 10^5, S \le 2\times 10^6, ty=1$，**16MB**

前面的在线算法的弊端在于，每次比较两个节点的dfs序的时候，需要用$O(\log n)$的时间在平衡树上求对应节点dfs序的排名。

可以考虑给每个节点附上一个权值，这个权值之间的相对大小关系对应着节点dfs序的相对大小关系。

为了解决权值精度的问题，可以使用后缀平衡树中的解决办法，用一棵重量平衡树（出题人使用了treap）维护节点dfs序，treap中每个节点表示一个区间(L,R)，这个节点的权值为$M=\frac{(L+R)}{2}$，其左儿子代表的区间为(L,M)，右儿子代表的区间是(M,R)，用线段树维护区间内的节点的权值的最值，每次往treap中加入一个节点的时候，如果某棵子树的权值不符合前面所说的，那么就暴力重构整棵子树的权值，由于相对大小没有变，所以没有必要在线段树上面修改，那么加入节点的期望复杂度是$O(n\log n)$的（详情参考陈立杰2013的国家集训队论文《重量平衡树和后缀平衡树在信息学竞赛中的应用》）。

这样做之后，修改的复杂度变成了$O(n\log n)$的，由于维护的东西变成了可以直接$O(1)$比较大小的实数，所以询问的复杂度变成了$O(S\log n)$.

时间复杂度$O((n+S)\log n)$，空间复杂度$O(n)$。