Day 8 模拟赛 题解

T1

没有问号的情况下，考虑如何线性判定。

考虑每两位当作一组，对于每组有如下两种操作：

1. 将两位依次压入栈中。
2. 将第一位与栈中全部元素合并后，再将第二位压入栈中。

可以发现栈中的情况可以看作是关于下一个压入元素的函数，即 $G[a,b](x)$，表示当 $x=0$ 时返回 a，$x=1$ 时返回 b。
假设当前栈中情况为 $G[a,b](x)$，考虑压入两位 c,d，容易根据上述两种操作合并出新的函数 $G[a',b'](x)$。
由于本质不同的函数只有 4 种，用 f_{i,j_0,j_1} 表示前 i 位能否得到 $G[j_0,j_1](x)$。

对于计数问题，dp 套 dp 即可，状态数是 $O(2^{2^2}n)$。

T2

不难发现的是，一个点的 SG 函数就是其子树内距离这个点最远的点到它的距离。一个棋子的 SG 值就是它所在点的 SG 值，整棵树的 SG 值就是所有棋子的 SG 值的异或和。

这样就不难做出比较小的数据和 $x=1$ 的部分分了。

考虑继续挖掘某个点的 SG 值的性质。对于一个点 x，我们先把它提为根，并对每个子树求出距离它最远的点的距离，记作集合 dis_x。假设当前的根是 y，那么 x 在以 y 为根的树的子树，就是整个连通块减去以 x 为根时包含 y 的那棵子树后剩下的连通块。这样的话，在以任意一个点为根时 x 子树内距离 x 最远的点到 x 的距离要么就是 dis_x 的最大值，要么就是 dis_x 的次大值。

可以发现的是，贡献 dis_x 的最大值的那个点，一定是直径的某个端点 u，并且从 u 走到 x 的路径，必定包含直径的中点。

我们设 f_x 为 dis_x 的最大值，g_x 为 dis_x 的次大值。在直径的中点为根的时候，所有点都会取到 g_x。当根为某个点 y 时，只有中点到 y 的路径上的点会取到 f_x，其它依旧是 g_x。

我们把 f,g 用矩阵维护，换根就相当于是矩阵翻转，而子树加、链加也可以直接维护，直接上树剖加线段树即可。

T3

不难发现题目的意思就是选择任意一个不为空的点集 V，使得 $\frac{e(V)}{|V|-1}$ 的值大于 lim。其中 $e(V)$ 是点集 V 构成的导出子图的边数。

移项可以得到 $e(V)-|V|lim>-lim$，不难发现这是一个最大权闭合子图的问题。其中一条边的权值是 1，一个点的权值是 $-lim$，如果选择了一条边那么就必须要选择它的两个端点。

但是如果直接求最大权闭合子图的话，我们会把什么都不选这种方案算进去，这是不合法的，所以需要处理这个问题。

考虑枚举一个点，并且强行选它。在这种情况下，这个点代表的边应该在网络流的图上被删掉，并且直接被算上 $-lim$ 的贡献。

这样不难得到一个暴力的做法，先枚举强行选的点，然后把这个点代表的边在网络流的图上删掉，然后跑一边最大权闭合子图看是否符合要求。但这个做法的时间复杂度很高。

考虑优化，可以发现，在枚举强行选的点时候，每次跑的网络流之间只有一条边的容量被修改，而其它部分是不变的，所以每次都重新跑一边网络流很不划算。考虑重复利用这个网络流的图，假设我们现在枚举的点是 i，那么在枚举到 $i+1$ 的时候，我们可以先把 $i+1$ 的流退掉，然后再重新增广。