Restoring Map 解题报告

题目大意

有一棵n个节点的树，对每个节点找出和它距离不超过2（包括自己）的点的集合。现在将这些集合打乱后给你，要求你复原出这棵树。如果有多解，输出任意一个。

数据范围

$2 \leq n \leq 1000$

解题过程

先特判掉$n = 2$的情况，下面假设$n \geq 3$。

考虑两个集合的交集：

当两个点的距离大于4时，显然集合交集为空集。

当两个点距离为4 时，如果路径为$u \to u_1 \to u_2 \to u_3 \to v$，则只有$u_2$与$u$、$v$距离都不超过2。

当两个点距离为3时，如果路径为$u \to u_1 \to u_2 \to v$，则只有u_1、u_2与u、v距离都不超过2，此时u_1，u_2均为非叶节点，且u_1，u_2间有边相连。

当两个点距离为2时，如果路径为$u \to u_1 \to v$，则至少有u、u_1、v三个点与u、v距离都不超过2。

当两个点距离为1时，所有与u有边相连的点和所有与v有边相连的点都与u、v距离都不超过2。当$n \geq 3$时，u、v的度数不可能都是1，此时至少有三个点与u、v距离都不超过2。

综上，当两个集合交集大小为2时，交集中的两个点必定都是非叶节点并且有边相连。如果u、v都是非叶节点并且有边相连，则必定存在一个不同于v的节点x与u有边相连，和一个不同于u的节点y与v有边相连，则只有u、v两点与x、y距离都不超过2。

枚举两个集合，判断交集大小是否为2。可以得到所有非叶节点之间的边。若有m条这样的边，就有$m + 1$个非叶节点。

情况 1：$m = 0$

此时任何两个点之间的距离都不超过2，所以输入的n个集合都是全集。任何一棵有一个非叶节点和$n - 1$个叶节点的树都满足要求。

情况 2、3：$m \geq 1$

此时可以找出哪些节点是非叶节点。注意到对于一个叶节点u，如果它和非叶节点v有边相连，则与u距离不超过2的节点集合就是与v距离不超过1的节点集合。

情况 2：$m = 1$

设两个非叶节点为u、v。找到任意一个不是全集的集合，将集合外的叶节点与u相连，集合内的叶节点与v相连，就是一个合法的解。

证明:此时所有节点和u、v的距离都不超过2，所以找到的这个集合必定是离某个叶节点距离不超过2的节点集合，也是离某个非叶节点距离不超过1的节点集合。

如果是离u距离不超过1的节点集合，则集合内的叶节点与u相连，集合外的叶节点与v相连。

如果是离v距离不超过1的节点集合，则集合内的叶节点与v相连，集合外的叶节点

与u相连。

而对于这两种答案，生成的输入数据相同，所以都是合法解。

情况 3：$m \geq 2$

此时只保留非叶节点会得到一棵点数大于等于3的树。

在新树中求出和每个节点距离不超过1的节点集合，这些集合两两不同。

如果一个输入的集合与非叶节点集合的交集等于新树中与节点u距离不超过1的节点集合，有下面几种情况:

这个输入的集合是与个叶节点距离不超过2的节点集合。则这个集合只可能是与节点u距离不超过1的节点集合。

这个输入的集合是与u距离不超过2的节点集合，则新树中所有节点与u距离不超过1。

这个输入的集合是与一个与u相邻的节点v距离不超过2的节点集合，则新树中v为叶节点。

对于后面两种情况，u在新树中必定是非叶节点，并且这个集合中必定包含与某个新树中的叶节点有边相连的原树中叶节点。

所以只要先处理新树中非叶节点，再处理新树中叶节点，就可以对每个非叶节点求出与它距离不超过1的节点集合（如果输入中不存在这样的集合，那么与节点u距离为1的节点均为非叶节点），从而求出原树。

由于集合求交可以用 bitset 优化，时间复杂度为$O(\frac{n^3}{w})$

Complete Compress 解题报告

题目大意

给一棵n个节点的树，有一些节点上有棋子，一次操作可以选择两个距离不小于2的棋子，将它们往中间移动一步，问能否将棋子移动到同一个节点上，如果可以，输出最小操作次数。

数据范围

$2 \leq n \leq 2000$

解题过程

先枚举最后棋子移动到了哪个节点上。

以棋子最后移动到的节点作为树根。

记$f[u]$为只考虑子树u时所有棋子的深度和在操作后能达到的最小值，$g[u]$为子树u中的所有棋子操作前的深度和，$size[u]$表示子树u中的棋子个数。$g[u]$和$size[u]$求法显然，考虑怎么求$f[u]$。

从$f[u] = g[u] = 0$开始，不断加入子树。在加入子树v时，令

$$f[u] \coloneqq \begin{cases} f[u] - (g[v] + size[v]) & (f[u] > g[v] + size[v]) \\ f[v] + size[v] - g[u] & (f[v] + size[v] > g[u]) \\ (g[u] + g[v] + size[v]) \bmod 2 & otherwise \end{cases}$$

然后更新$g[u]$。

下面证明这么做的正确性。

定理1：如果将一个节点上的棋子移动到它的父节点上，按上面的方式求出的f的值不会小于移动前求出的f的值减1。

证明：

使用数学归纳法。

（1）在子树u只有一个节点时，显然成立。

（2）假设在加入子树v之前成立。

（记移动后的f,g为f',g'，加入子树v前的$f[u]$为$f_0[u]$）

若移动的棋子不在子树v中，则$f'[v]=f[v]$，$g'[v]=g[v]$，$size'[v]=size[v]$，$g'_0[u]=g_0[u]-1$，$f'_0[u]\geq f_0[u]-1$。

当$f_0[u]>g[v]+size[v]+1$时，$f'_0[u]\geq f_0[u]-1>g[v]+size[v]$，$f'[u]=f'_0[u]-(g[v]+size[v])\geq f_0[u]-(g[v]+size[v])-1=f[u]-1$。

当$f[v]+size[v]>g_0[u]$时，$g'_0[u]=g_0[u]-1<g_0[u]<f[v]+size[v]$，$f'[u]=f[v]+size[v]-g'_0[u]=f[v]+size[v]-g_0[u]+1=f[u]+1>f[u]-1$。

对于其它情况，显然$f[u]\leq 1$，$f'[u]\geq 0\geq f[u]-1$。

当移动的棋子在子树v中（包括节点v）时，有$f'_0[u]=f_0[u]$，$g'_0[u]=g_0[u]$，$f'[v]+size'[v]\geq f[v]+size[v]-1$，$g'[v]+size'[v]=g[v]+size[v]-1$。（如果是将节点$v$上的棋子移动到节点$u$，注意到根节点的棋子数量不会影响$f$和$g$的值，所以这个仍然成立）。同理可证$f'[u]\geq f[u]-1$。

（3）综合（1）（2），可证定理成立。

定理 1 证完。

定理2：如果进行一次操作，按上面的方式求出的f的值不会减小。

证明：

使用数学归纳法。

（1）在子树u只有一个节点时，显然成立。

（2）假设在加入子树v之前成立。

（记操作后的f,g为f',g'，加入子树v前的$f[u]$为$f_0[u]$）

若操作的两个棋子都不在子树v中，则$f'[v]=f[v]$，$g'[v]=g[v]$，$size'[v]=size[v]$，$g'_0[u]\leq g_0[u]$，$f'_0[u]\geq f_0[u]$。

当$f_0[u]>g[v]+size[v]$时，$f'_0[u]\geq f_0[u]>g[v]+size[v]$，$f'[u]=f'_0[u]-(g[v]+size[v])\geq f_0[u]-(g[v]+size[v])=f[u]$。

当$f[v]+size[v]>g_0[u]$时，$g'_0[u]\leq g_0[u]<f[v]+size[v]$，$f'[u]=f[v]+size[v]-g'_0[u]\geq f[v]+size[v]-g_0[u]=f[u]$。

对于其它情况，有$f[u]=(g_0[u]+g[v]+size[v])\ mod\ 2=g[u]\ mod\ 2=g'[u]\ mod\ 2=f'[u]\ mod\ 2\ \leq f'[u]$（可以证明$g[u]\ mod\ 2=f[u]\ mod\ 2$）。

若操作的两个棋子都在子树v中（包括操作节点u的棋子和子树v中的棋子），则$f'[v]+size'[v]\geq f[v]+size[v]$，$g'[v]+size'[v]\leq g[v]+size[v]$，$g'_0[u]=g_0[u]$，$f'_0[u]=f_0[u]$。同理可证$f'[u]\geq f[u]$。

若操作的两个棋子一个在子树v中，一个不在（不包括操作节点u的棋子和子树v中的棋子），则$f'[v]+size'[v]\geq f[v]+size[v]-1$，$g'[v]+size'[v]=g[v]+size[v]-1$，$f'_0[u]\geq f_0[u]-1$，$g'_0[u]=g_0[u]-1$。（使用定理 1）

当$f_0[u]>g[v]+size[v]$时，$f'_0[u]\geq f_0[u]-1>g[v]+size[v]-1=g'[v]+size'[v]$，$f'[u]=f'_0[u]-(g'[v]+size'[v])\geq f_0[u]-1-(g[v]+size[v]+1)=f_0[u]-(g[v]+size[v])=f[u]$。

同理，当$f[v]+size[v]>g_0[u]$时，$f'[u]\geq f[u]$。

对于其它情况，有$f[u]=(g_0[u]+g[v]+size[v])\ mod\ 2=g[u]\ mod\ 2=g'[u]\ mod\ 2=f'[u]\ mod\ 2\ \leq f'[u]$。

（3）综合（1）（2），可证定理成立。

定理 2 证完。

定理3：在只考虑子树u时，可以通过$\frac{g[u]-f[u]}{2}$次操作将所有棋子的深度和降至$f[u]$，且不可能降到更低。

证明：

使用数学归纳证明定理前半部分。

（1）在子树u只有一个节点时，显然成立。

（2）假设在加入子树v之前成立。

(记操作后的f,g为f',g'，加入子树v前的$f[u]$为$f_0[u]$)

当$f_0[u]>g[v]+size[v]$时，可以操作子树v之外的棋子直至$g_0'[u]=f_0[u]$，然后再不断操作子树v之外的一个棋子和子树v内的一个棋子。

当$f[v]+size[v]>g_0[u]$时，可以操作子树v之内的棋子直至$g'[v]=f[v]$，然后再不断操作子树v之外的一个棋子和子树v内的一个棋子。

对于其它情况，令$x=\min(g_0[u],g[v]+size[v])$，则有$f_0[u]\leq x\leq g_0[u]$，$f[v]+size[v]\leq x\leq g[v]+size[v]$，则通过操作将$g_0[u]$降至$f_0[u]$的过程中，必定有一个时刻$g_0'[u]\in\{x-1,x\}$，通过操作将$g[v]$降至$f[v]$的过程中，必定有一个时刻$g'[v]+size'[v]\in\{x-1,x\}$。操作到这个时刻之后再不断操作子树$v$之外的一个棋子和子树$v$内的一个棋子，可以使$g'[u]\leq 1$。而$g'[u]\ mod\ 2=g[u]\ mod\ 2$，所以此时$g'[u]=g[u]\ mod\ 2=f[u]$。

（3）综合（1）（2），可证：在只考虑子树u时，可以通过$\frac{g[u]-f[u]}{2}$次操作将所有棋子的深度和降至$f[u]$。

由于$g'[u]\geq f'[u]\geq f[u]$（根据定理 2 得出），所以深度和不可能降到更低。

定理 3 证完

根据定理3，当$f[root]=0$时，可以通过$\frac{g[root]}{2}$次操作将所有棋子移动到$root$上，否则不能将所有棋子移动到$root$上。

单次计算所需时间为$O(n)$，需要枚举n个节点分别计算，总时间复杂度$O(n^2)$。

证完。

Shopping 解题报告

题目大意

在一个数轴上有一列火车在坐标0和坐标L间以每秒钟单位1的速度反复移动。数轴上有n个购物中心，第i个购物中心坐标在x_i需要连续购物至少t_i时间，问从原点开始坐火车到每个购物中心购物一次然后回到原点需要多少时间。

数据范围

$1 \le n \le 300000$
$1 \le L \le 10^9$
$0 < x_1 < x_2 < \cdots < x_n < L \le 10^9$

解题过程

考虑计算火车到两端的最少次数，则答案为这个最少次数$\times L$。

由于$t_i > 2L$时火车到两端的次数为将t_i视为$t_i - 2L$时到两端的次数+2，所以下面只考虑$t_i \le 2L$的情况。

我们将 Yui 进入一个车站（进入一个购物中心购物或者到达端点然后返回）的方式分成三类:

方式（1）：从左边进入，从左边离开。

方式（2）：从右边进入，从右边离开（包括回到原点不离开）。

方式（3）：从左边进入，从右边离开或者从右边进入，从左边离开。

考虑如果给出进入每个车站的方式，怎么判断能否做到按给出的进入的方式进入每个车站：

如果是按方式（1）进入，在这个位置记录上-1；如果是按方式（2）进入，记录上1；按方式（3）进入则记录0。则能够按要求进入每个车站，当且仅当所有位置记录的数的和为0，且所有非空、非全集的前缀和为正。

证明：

对于一个位置，显然 Yui 从左边经过这个位置的次数等于这个位置左边的数的和。

如果所有数的和非零，则经过一个坐标L右边的位置的次数非零，这是不可能的。

如果存在一个非空，非全集的前缀和非正，那么存在一个位置在原点右侧，从左边经过它的次数非正，并且在这个位置右侧还有需要进入的车站，这是不可能的。

而如果所有位置记录的数的和为0，且所有非空、非全集的前缀和为正，则将1看成左括号，-1看成右括号（0只需要 Yui 的路线经过这个车站即可），对应一个合法的括号序列，且最后一个右括号和第一个左括号对应。那么原点→第一个右括号→第一个右括号对应的左括号→第二个右括号→第二个右括号对应的左括号→ … →最后一个右括号→最后一个右括号对应的左括号（原点）就是合法的路径。

证完

下面考虑如何解决原题：

显然能上火车的时候一定上火车。由于$t_i \le 2L$，所以在购物中心购物时，火车到达端点的次数（下称“代价”）只会是1或2。

如果从左边进入和从右边进入代价都是1，那么可以自由决定以方式（1）进入还是以方式（2）进入。

如果从左边进入代价是1，从右边进入代价是2，那么显然从左边进入不会更劣，也就是只能以方式（1）进入。

如果从左边进入代价是2，从右边进入代价是1，那么显然从右边进入不会更劣，也就是只能以方式（2）进入。

如果从左边进入和从右边进入代价都是2，那么只能以方式（3）进入。

现在问题转换成了：给定一个包含问号的括号序列，要求确定问号的取值，然后在两边添加若干个括号，使得括号序列合法，且第一个左括号和最后一个右括号对应，要求括号数量最少。

显然前面一部分问号变成左括号，后面一部分变成右括号。枚举那些变成左括号，可以通过预处理前后缀和做到$O(1)$判断添加的括号的数量。总复杂度$O(n)$。