集训队作业题选讲

成都七中 蒋明润

2021.2.2

目 录

1 IOI相关

IOI的赛制特点

- 多次提交
- 实时反馈
- 没有罚时
- 有部分分

一点经验

- 打好数学基础
- 多看经典书籍
- 多刷题

2

IOI2020D1T1

plants

- **题目大意**
  - 在一个环上有n棵高度互不相同的植物，现在对每一棵植物给出它后面$k-1$棵植物有多少棵比它高，然后q次询问两棵植物高度的大小关系（或确认无法通过已知信息判定大小关系）
- **数据范围：**$2 \le k \le n \le 200000$

$$q \le 200000$$

保证输入合法

plants

- **解题过程**
  - 引理1：通过已知信息可以唯一确定任意两棵距离小于k的植物高度的大小关系
  - 如果引理1正确，那么可以对于任意两棵距离小于k的植物a,b，如果a比b高，就从a向b连一条有向边。不难证明这样会形成一个DAG，而任意一个合法的排列都对应这个DAG的一个拓扑序。两棵植物可以确定高度的大小关系当且仅当它们在DAG上可达。

plants

- **解题过程**
  - 先考虑如何找到一个合法的排列。
  - 使用类似拓扑排序的方法，这需要不断寻找“入度为0的点”，也就是找到一棵植物满足它高于所有与它距离小于 k 且未被删除的植物。

plants

- **解题过程**
  - 对每棵未被删除植物维护出它后面的$k-1$棵植物中有多少植物未被删除且比它高，那么一棵植物满足它高于所有与它距离小于k且未被删除的植物当且仅当这棵植物上维护的数是0并且它前面的$k-1$棵植物中未被删除的植物上维护的数都不是0。而删除一棵植物时，它前面的$k-1$棵植物上维护的数会减1。

plants

- **解题过程**
  - 这样问题转化成区间加，查询一个 0 （最小值）满足它前面的$k-1$个数都不是0（如果找不到就返回最前面的0），这可以用线段树维护，时间复杂度为$O(n\log n)$
  - 通过这个过程我们也证明了引理1的正确性：每次删除一棵植物时都可以确定它和所有和它距离小于k的植物的大小关系，而植物在最终会删完。

plants

- **解题过程**
  - 下面考虑如何判断可达性。
  - 对每一棵植物x，可以用线段树求出拓扑序在x后的植物中第一棵在x前面$k-1$个位置中的树和第一棵在x后面$k-1$个位置中的树。记作$left[x]$和$right[x]$。

plants

- **解题过程**
  - 如果要查询x能否到达y，可以从x开始不断跳$left[x]$直到继续跳拓扑序就会在y之后（并用相同的方法从x不断跳$right[x]$），不难证明x能到达y当且仅当通过这个不断跳$left[x]$和不断跳$right[x]$的过程最后得到的区间包含y（不过需要注意这个过程有可能会在环上绕好几圈，此时得到的区间应当是全集）。而这个不断跳$left[x]$和不断跳$right[x]$的过程可以通过倍增优化，时间复杂度为$O(n\log n+q\log n)$。

3

作业题选讲

题目1 01 on Tree

- **试题来源**
  - 本题来源于2020国集作业第89题，原题地址为：https://atcoder.jp/contests/agc023/tasks/agc023_f

题目1 01 on Tree

- **题目大意**
  - 给一棵n个节点的有根树，树上每个节点有一个数字，0或1，要求将这些节点排成一行，祖先节点必须在后代节点的前面，使得将节点上的数按顺序写下后的逆序对最少，输出逆序对数。
- **数据范围：** $1 \leq n \leq 2\times10^5$

题目1 01 on Tree

- **解题过程**
  - 思路：直接解决这个不容易，可以考虑解决一个更强的问题：每个节点上写的是一个非空01序列。

题目1 01 on Tree

- **解题过程**
  - 设$a[i]$为节点i上的序列中0的个数，$b[i]$为节点i上的序列中1的个数。
  - 设排成的序列为p，考虑交换$p[i]$和$p[i+1]$：
  - 此时会有$b\left[p[i]\right] \times a[p[i+1]]$对数从逆序对变成不是逆序对，有$b\left[p[i+1]\right] \times a[p[i]]$对数从不是逆序对变成逆序对。
  - 所以当$b\left[p[i+1]\right] \times a[p[i]] \le b\left[p[i]\right] \times a[p[i+1]]$，也就是$b\left[p[i+1]\right] / a[p[i+1]] \le b[p[i]] / a[p[i]]$（若分母为0，则认为分数的值为无限大），且交换后仍然是合法的序列时，交换后答案不会变劣。

题目1 01 on Tree

- **解题过程**
  - 考虑满足$b[i]/a[i]$最小的非根节点i，我们可以通过交换将节点i不断向前移动直到移动到i的父节点后面一个位置，且移动过程中答案不会变劣。所以必定存在一个最优解，满足i在i的父节点后面一个位置。所以可以将i和i的父节点合并，合并后的01序列为两个01序列拼在一起。反复执行该操作直到树只剩下根节点，此时根节点上的01序列就是最后的01序列。
  - 用堆维护$b[i]/a[i]$最小的非根节点，用并查集维护合并之后每个节点的父节点，可以做到时间复杂度$O(nlogn)$

题目2 Painting Edges

- **试题来源**
  - 本题来源于2020国集作业第123题，原题地址为：https://codeforces.com/contest/576/problem/E

题目2 Painting Edges

- **题目大意**
  - 给一张n个点m条边的无向图，初始时每条边没有颜色。q次操作，每次操作将一条边染成k种颜色中的一种。如果一次操作后某种颜色的边构成的子图不是二分图，该操作不会被执行。判断每次操作是否被执行。
- **数据范围**：$n, m, q \leq 5\times10^5$，$k \leq 50$
- **时间限制**：$6\ s$
- **空间限制**：$600M$

题目2 Painting Edges

- **解题过程**
  - 如果考虑在线情况，需要动态判定一张图加上一条边之后是否是二分图，而且有加边和删边（把一条边染成别的颜色）操作，难以维护。所以首先考虑离线。

题目2 Painting Edges

- **解题过程**
  - 1.考虑离线怎么动态判定一张图加上一条边之后是否是二分图：
  - 考虑线段树分治。我们可以求出每条边出现的时间段，然后定位到线段树上的$O(logn)$个节点。然后在线段树上dfs，访问到一个节点时将边加入，退出时撤销。这样问题转换成了加边、撤销、查询。而询问加入一条边后是否是二分图可以用带权并查集维护。定位到$O(logn)$个节点后有$O(nlogn)$条边，可撤销并查集的时间复杂度是$O(logn)$，总时间复杂度$O(n\ log^2 n)$（此处认为n, q同阶）

题目2 Painting Edges

- **解题过程**
  - 2.但是离线做法在本题中会有一个问题：我们无法预先知道每个操作是否执行，也就无法知道每条边的出现时间。下面解决这个问题：
  - 对于操作执行的情况，我们可以认为是删去原来的边然后加入新的颜色的边，而对于操作不执行的情况，我们可以认为是删去原来的边然后加入颜色和原来的边相同（可能是没有颜色）的边。这样我们仍然可以知道每条边出现的时间段，只是不知道每条边的颜色。所以仍然可以定位到线段树上的$O(logn)$个节点。而在线段树上dfs的时候，确定一条边的颜色的时间在加入这条边之前，所以不知道颜色也不是问题。

题目2 Painting Edges

- **解题过程**
  - 时间复杂度O($nlog^2n + nk$)(nk是初始化k个并查集)，空间复杂度O($nlogn + nk$)。
  - 另外，也可以使用LCT维护删除时间最大生成树的方法维护，时间复杂度为O($nlogn + nk$)，不过常数较大，实际表现不一定更优。

题目3 Restoring Map

- **试题来源**
  - 本题来源于2020国集作业第36题，原题地址为：https://codeforces.com/contest/566/problem/E

题目3 Restoring Map

- **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：**$2 \le n \le 1000$

题目3 Restoring Map

- **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。

题目3 Restoring Map

- **解题过程**
  - 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

题目3 Restoring Map

- **解题过程**
  - 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。
  - 对于剩下的情况，可以找出哪些节点是非叶节点，同时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。

题目3 Restoring Map

- **解题过程**
  - 情况2：$m=1$
    - 设两个非叶节点为u、v。
    - 此时所有节点和u、v的距离都不超过2，所以任意一个不是全集的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。

题目3 Restoring Map

- **解题过程**
  - 情况2：$m = 1$
    - 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。

题目3 Restoring Map

- **解题过程**
  - 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。

题目3 Restoring Map

- **解题过程**
  - 情况3：$m \geq 2$
    - 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一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相邻的非叶节点v距离不超过2的节点集合，则新树中v为叶节点。

题目3 Restoring Map

- **解题过程**
  - 情况3：$m \geq 2$
    - 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叶节点，再处理新树中非叶节点，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。

题目3 Restoring Map

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

题目4 Complete Compress

- **试题来源**
  - 本题来源于2020国集作业第37题，原题地址为：https://atcoder.jp/contests/agc034/tasks/agc034_e

题目4 Complete Compress

- **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**：$2 \le n \le 2000$

题目4 Complete Compress

- **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$f[u]$。

题目4 Complete Compress

- **解题过程**
  - 从$f[u]=g[u]=0$开始，不断加入子树。在加入子树v时，令
  - $f[u]:=$
    $$\begin{cases} f[u]-(g[v]+size[v])\ (f[u]>g[v]+size[v]) \\ f[v]+size[v]-g[u]\ (f[v]+size[v]>g[u]) \\ (g[u]+g[v]+size[v])\ mod\ 2\ otherwise \end{cases}$$
  - 然后更新$g[u]$。

题目4 Complete Compress

- **解题过程**
  - 可以考虑用数学归纳法依次证明以下三个定理，具体证明这里略去
    - 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。
    - 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。
    - 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。

题目4 Complete Compress

- **解题过程**
  - 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$O(n^2)$。

题目5 Shopping

- **试题来源**
  - 本题来源于2020国集作业第38题，原题地址为：https://atcoder.jp/contests/agc022/tasks/agc022_d

题目5 Shopping

- **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：** $1 \le n \le 300000$

$$1 \le L \le 10^9$$
$$1 \le t_i \le 10^9$$

题目5 Shopping

- **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$t_i \leq 2L$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。

题目5 Shopping

- **解题过程**
  - 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。

题目5 Shopping

- **解题过程**
  - 证明如下：
    - 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。

题目5 Shopping

- **解题过程**
  - 证明（续）：
    - 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。

题目5 Shopping

- **解题过程**
  - 下面考虑如何解决原问题
  - 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。

题目5 Shopping

- **解题过程**
  - 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。枚举哪些变成左括号，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$O(n)$。

题目6 Donation

- **试题来源**
  - 本题来源于2020国集作业第152题，原题地址为：https://atcoder.jp/contests/arc098/tasks/arc098_d

题目6 Donation

- **题目大意**
  - 给一张n个点，m条边的无向图，每个点上有两个权值a_i和b_i，你可以从任意一个点出发，然后沿着图中任意边行走，当你进入一个点u（包括选择从点u出发）时，需保证此时的钱数不小于a_u，当你位于点u时，可以消耗b_u的钱捐赠一次（不能使剩余钱数为负）。问初始时需要多少钱才能在每个点捐赠一次。
- **数据范围：** $1 \le n \le 100000$

$$n - 1 \le m \le 100000$$

$$1 \le a_i, b_i \le 10^9$$

题目6 Donation

- **解题过程**
  - 正向考虑这个过程较难处理（虽然也可以处理），我们逆向考虑这个过程。
  - 问题等价于：你可以以任意非负初始钱数从任意一个点出发，然后沿着图中任意边行走，当你离开一个点u（包括最终停止在点u）时，需保证此时的钱数不小于a_u，当你位于点u时，可以“捐赠”一次使钱数增加b_u，要求给每个点“捐赠”一次，最终的钱数最少。

题目6 Donation

- **解题过程**
  - 每一个点都会捐赠一次，所以最终钱数最少等价于初始钱数最少。
  - 显然对于每一个点会在第一次访问这个点时“捐赠”。
  - 令$c_u = \max(a_u - b_u, 0)$，则进入点u时只需保证钱数不小于c_u（即“捐赠”一次后不小于a_u）。

题目6 Donation

- **解题过程**
  - 按照c_u从小到大的顺序加入每一个点，用并查集维护连通块，可以得到一个树形结构，子树u代表加入u时形成的连通块。
  - 不难证明在确定初始位置和钱数之后，可以以任意顺序访问可以访问的节点。

题目6 Donation

- **解题过程**
  - 当访问到点u时，子树u中的所有点的c值都不超过c_u，可以访问。
  - 访问完子树u后，能到达的未访问点中c值最小的点为u的父节点f，此时访问f必然可行，否则无解。
  - 访问节点f后可以访问子树f，然后访问f的父节点，以此类推

题目6 Donation

- **解题过程**
  - $dp[u]$表示访问子树u需要的最少钱数，转移时枚举访问u之前先访问了哪个子树。
  - DP的时间复杂度为$O(n)$，时间复杂度瓶颈为排序的$O(nlogn)$

题目7 Mr. Kitayuta vs. Bamboos

- **试题来源**
  - 本题来源于2020国集作业第150题，原题地址为：https://codeforces.com/contest/506/problem/C

题目7 Mr. Kitayuta vs. Bamboos

- **题目大意**
  - 有n棵竹子，初始高度为h_i，每天结束后高度会增加a_i，每天可以做k次操作，每次操作可以选择一棵竹子将高度减少p（高度小于p则减至0，不会变成负数），最小化m天后竹子高度的最大值
- **数据范围**：$1 \leq n \leq 100000$

$$1 \leq m \leq 5000$$
$$1 \leq k \leq 10$$
$$1 \leq p \leq 10^9$$
$$0 \leq h_i \leq 10^9$$
$$1 \leq a_i \leq 10^9$$

题目7 Mr. Kitayuta vs. Bamboos

- **解题过程**
  - 首先二分答案，问题转化为判断能否使m天后所有竹子高度均不超过X。
  - 逆向考虑这个过程，记h'表示如果在原问题中此时竹子的高度超过h'，则竹子最终的高度会超过X
  - 则在逆向过程中，h'的初始值为X，在每天开始时h'会减少a_i，每天可以做k次操作，每次操作可以选择一棵竹子将h'增加p（不能在h'减小至负数后增加），要求m天后h'不小于h_i

题目7 Mr. Kitayuta vs. Bamboos

- **解题过程**
  - 最优决策为选择（如果不操作）h'减小至不合法（减小至负数或者过程结束后小于h_i）的时间最早的竹子进行操作。
  - 可以用堆维护，时间复杂度为$O((n+km)logn)$，算上二分的时间复杂度后为$O((n+km)logn\ logH)$

题目8 Encoding Subsets

- **试题来源**
  - 本题来源于2020国集作业第104题，原题地址为：https://atcoder.jp/contests/agc020/tasks/agc020_e

题目8 Encoding Subsets

- **题目大意**
  - 对于一个01串，我们可以按下面的方式编码：
    - 0和1分别可以被编码为0和1
    - 若A可以被编码为P，B可以被编码为Q，则AB可以被编码为PQ
    - 若A可以被编码为P，$\mathrm{K} \geq 2$，则A重复K次可以被编码为$(P \times K)$
  - 定义01串A是B的子集当且仅当A和B长度相同且所有满足A_i为1的位置均有B_i为1
  - 给定一个01串S，求S的所有子集的编码方案数之和
- **数据范围**：$1 \leq |S| \leq 100$

题目8 Encoding Subsets

- **解题过程**
  - 令$dp(S)$表示S的所有子集的编码方案数之和。
  - 考虑S的子集A的一种编码方案的结尾，有两种情况：
  - 1.以单独一个字符结尾，则该字符为A的最后一个字符，A的剩余部分必须是S的相应位置的子集，有$dp(S[1..|S|-1])$种编码方案（若S的结尾为1，则A的结尾为0或1 均可，方案数需乘2）。

题目8 Encoding Subsets

- **解题过程**
  - 2.以$(P\times K)$的形式结尾
  - 设P为01串B的编码，编码方案剩余部分为C的编码，则A为$CB\ldots B$（B 重复K次）。
  - 枚举$|B|$和K，可确定C是$S[1..|C|]$的子集，B是相应的K个01串的子集。
  - 定义长度相同的01串的交集为对应位置取min后的01串，则B是这K个01串的交集T的子集。
  - 编码方案数为$dp(S[1..|C|])\times dp(T)$

题目8 Encoding Subsets

- **解题过程**
  - 这样做看上去状态数是$O(2^{|S|})$，不过实际有用的状态很少，下面大致分析一下状态数。
  - 有用的状态必然是从S开始不断进行下面两种操作得到的：
    - 取一个子串
    - $K-fold$这个字符串，即将这个字符串分成K个长度相同的串然后求交集

题目8 Encoding Subsets

- **解题过程**
  - 下面分析进行两轮操作后得到的串的数量（这里只分析$2-fold$，K更大时分析方法类似）
  - 设第一轮操作后得到的串T为$S[L_1+1..L_1+len_1]$和$S[L_1+len_1+1..L_1+2\times len_1]$的交集（即$S[L_1+1..L_1+2\times len_1]$ $2-fold$的结果），第二轮操作后得到的串为T$[L_2+1..L_2+len_2]$和$T[L_2+len_2+1..L_2+2\times len_2]$的交集。

题目8 Encoding Subsets

- **解题过程**
  - 则两轮操作后得到的串为$S[L_1+L_2+1..L_1+L_2+len_2]$，$S[L_1+L_2+len_2+1..L_1+L_2+2\times len_2]$，$S[L_1+len_1+L_2+1..L_1+len_1+L_2+len_2]$，$S[L_1+len_1+L_2+len_2+1..L_1+len_1+L_2+2\times len_2]$这四个串的交集，注意到它们只和$L_1+L_2$，$len_1$，$len_2$有关，所以数量为$O(n^3)$

题目8 Encoding Subsets

- **解题过程**
  - 而如果进行超过两轮的操作，得到的串的长度会小于等于$\frac{|S|}{8}$，数量为$O(2^{\frac{|S|}{8}})$。这样就可以得到状态数的一个上界
  - 如果继续考虑三轮，四轮操作之后的串的数量，可以得到理论上更优的奇怪复杂度。

题目9 Boolean Function

- **试题来源**
  - 本题来源于2020国集作业第99题，原题地址为：https://codeforces.com/contest/582/problem/E

题目9 Boolean Function

- **题目大意**
  - 给你一个逻辑表达式s，包含A, B, C, D四个变量和&,|两个运算符，任意一个运算符两侧的子表达式都会加上括号。现在有一些运算符和变量消失了，用?代替。已知n种不同的变量取值下这个表达式的值，求有多少种可能的表达式。
- **数据范围：** $1 \le |s| \le 500$

  $0 \le n \le 16$

题目9 Boolean Function

- **解题过程**
  - 将每一种变量取值下表达式s的值记作一个n位二进制数$g(s)$，显然$g((s1)|(s2)) = g(s1)|g(s2)$，$g((s1)\&(s2)) = g(s1)\&g(s2)$。
  - 建出输入的表达式s的表达式树，$dp[u][x]$表示节点u代表的子表达式t有多少种方案满足$g(t) = x$。
  - 转移时先考虑枚举两个子节点的g，用$dp[ls][x]\times dp[rs][y]$更新$dp[u][x|y]$和$dp[u][x\&y]$。

题目9 Boolean Function

- **解题过程**
  - 此时时间复杂度为$O(4^n|s|)$，并不能通过。
  - 注意到这里的DP转移是一个或卷积或者一个与卷积的形式，可以使用fmt优化至$O(n\ 2^n)$。
  - 总时间复杂度为$O(n\ 2^n|s|)$

题目10 Construction of a tree

- **试题来源**
  - 本题来源于2020国集作业第145题，原题地址为：https://atcoder.jp/contests/agc029/tasks/agc029_f

题目10 Construction of a tree

- **题目大意**
  - 给出$n-1$个集合，每个集合的元素都是1到n中的正整数。要求从每个集合中选出两个元素u_i和v_i，在u_i和v_i两点间连边，使最终形成的图是一棵树，或者判断无解。
- **数据范围：**$1 \le n \le 100000$

所有集合的大小之和不超过200000

题目10 Construction of a tree

- **解题过程**
  - 考虑建立一张二分图，左侧代表每一个点，右侧代表每一个集合，每个集合向它的每一个元素所对应的点连边。
  - 首先假设存在一组解，考虑任意选取一个点为根，每一个非根节点与连向它父亲的边匹配。这样可以得到一个大小为$n-1$的匹配，只有根节点是未匹配点。
  - 根节点是任意选取的，所以左侧每一个点均可以作为未匹配点。

题目10 Construction of a tree

- **解题过程**
  - 使用dinic求出这个二分图的最大匹配，则左侧一个节点可以作为未匹配点当且仅当在残量网络中可以从源点到达这个点。
  - 若最大匹配不为$n-1$，或者残量网络中从源点无法到达所有左侧节点，则无解。
  - 考虑在残量网络中求出以源点为根的bfs树，则每一个已匹配的左侧节点u的父节点为与它匹配的右侧节点v，而这个右侧节点的父节点是另外一个与它有边相连的左侧节点w。

题目10 Construction of a tree

- **解题过程**
  - u和w均为集合v的元素，可以在集合v中选出u和w相连。
  - 显然这样会得到一棵树，满足要求。
  - 时间复杂度为$O(m\sqrt{n})$（ m 为所有集合的大小之和）

题目10 Construction of a tree

- **扩展**
  - 本题的解题方法与下题比较类似：
  - 2020国集作业第45题，原题地址为：https://codeforces.com/contest/611/problem/H
- **题目大意**
  - 有一棵n个节点的树，节点编号从1到n，现在你只知道每条边两端的节点编号在十进制下的位数，要求构造出任意一个符合条件的树，或者判断无解。
- **数据范围**：$1 \leq n \leq 200000$

中国计算机学会
China Computer Federation

谢谢您的聆听

THANKS YOUR LISTENING