New Year and Forgotten Tree(CF611H)

题目大意

一棵n个点的树，编号从1到n，你现在只知道每条边两个端点的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

$2 \leq n \leq 2 \times 10^5$

解题过程

位数相同的点染上相同的颜色，每一种颜色的点选一个大点，其他的是小点。

如果存在合法的树，一定有一种情况是大点连成一个连通块，小点连向某一个大点。

证明：

随便找一棵合法的树

如果存在小点--小点的边

1、如果这条边在两个小点同颜色的大点之间的路径上，可以把这条边接在这两个大点上。

2、否则这两个大点一定在同某一个小点的子树内，或者是两个小点颜色相同，对应的大点在其中一个的子树内，可以将另一边的小点连接到一个大点上。

把所有小点-小点的边去掉之后，如果大点不连通，那么一定是一个小点连向了多个大点

断掉小点连向的其中两个大点的边（这里一共分成3个连通块），一定有一个大点跟与小点同颜色的大点不在同一个连通块内，连接着两个大点，再将另一个大点连向原来的小点。

重复这个操作，直到小点只连向一个大点。

这样调整出来的树一定符合条件。

枚举大点之间的连通方式，剩下的边一定是(大--小)的形式，考虑是哪边的大点连哪边的小点，可以用网络流判断是否有解并构造出一组解，建图：原点连向每一种边，流量为边的数量，每一种边给对应颜色的点连流量无穷大的边，每种颜色再连向汇点，流量为点数，全部边满流则代表有解。可以根据网络流的解去构造答案。

Negative Cycle(AGC036D)

题目大意

一张n个点的有向图，编号从0到n-1，一开始有$n-1$条连接$(i \to i+1)$的边$(0 \leq i < n-1)$。现在对于每一对有序对(i,j)连一条边，如果i<j，则边权为-1，否则边权是1。对于每个边权不为0的边$(i \to j)$，可以花费$A_{i,j}$的代价删掉，求最小的删边代价使得图中没有负环。

数据范围

$3 \leq n \leq 500, 1 \leq A_{i,j} \leq 10^9$

解题过程

由于没有负环，考虑0到点i的最短路的长度p_i，有以下条件成立

> 任意边$(i\to j)$,存在$p_j\le p_i+weight$。
>
> 任意点i,存在从0到i的一条链使得每条边两端的最短路差是边权。

因为一开始的边权为0的边永远存在，所以$p_{i+1}\le p_i$

由于边权是±1，所以p_i的值域一定是连续的整数，又因为$p_{i+1}\le p_i$，所以有$p_i\le p_{i+1}+1$。

设$q_i=p_i-p_{i+1}$，$0\le q_i\le 1$，因为p_i表示0到i的最短路，所以有以下条件成立

> 对于一条$(i\to j),i<j$的边，$p_i-p_j\ge 1$即至少跨过一个$q=1$的位置。
>
> 当$i>j$时，有$p_j-p_i\le 1$，即这条边最多跨过一个$q=1$的位置。
>
> 对于每个$q_i=1$的位置，一定有一条边$(x\to i+1),x\le i,p_x=p_i$存在。这个条件与第一个不矛盾，因为这条边跨过了$q_i=1$这个位置。

如果确定了所有$q=1$的位置（即确定0号点到每个点的最短路），由于所有边的代价都是正数，最优解必然是保留所有符合上面条件的边，而对于所有$q_i=1$的位置，总有边$(i\to i+1)$符合条件。

设$dp[i][j]$表示上两个$q_x=1$的位置是i和j的时候，端点在i之前的符合条件的边的代价之和的最大值，转移就是枚举下一个$q=1$的位置。

ABBreviate(AGC027E)

题目大意

有一个只包含字符`a`,`b`的字符串s,可以以任意顺序对其做以下两种操作任意多次:

1、将字符串中的子串`aa`替换成`b`

2、将字符串中的子串`bb`替换成`a`

一个串s通过一系列操作生成多少个字符串？答案对10^9+7取模。

数据范围

$1\le|s|\le 10^5$

解题过程

一个长度大于1的串能被缩成一个字符当且仅当以下条件成立

1、a的个数+b的个数*2模3（记这个值为串s的权值）不是0

2、存在两个相邻的相同的字符

因为把a看成1，把b看成2时不论怎么缩和模3都是不变的，当存在两个字符相邻的时候一定有一种方法使得一次操作之后仍有相邻的相同的字符。

对于一个从s通过一系列操作得来的串t，那么t的每一个字符都可以通过以下方法对应到s的一个连续子串：从上一个字符对应的子串后面开始找到下一段最短的能缩成当前字符的一段。

举个例子：s=aaababbababbaaaba，t=abaa

t可以表示成|a|aa|babb|bababb|aaaba

两个分隔符之间的串可以缩成t中对应的字符，最后的aaaba是剩下的。

这样t就可以对应成s的前缀的划分，所有串t都能这样表示，证明如下：

> 假设t对应一种s的划分。
>
> 这里假设划分中的第一个字符对应的子串不是最短的（否则可以把第一个字符和对应的子串去掉），那么：
>
> |AB|C|（`|`是分隔符，即划分中第一个字符对应的是AB，而A是最短的能组成同样字符的前缀，C是第二个字符的对应的串）这样就有B的权值为0
>
> 如果将B并入下一个划分，即变成|A|BC|这样BC是满足第一个条件的，如果不满足第二个条件，那么一定是类似`abababababa...`的形式，此时C一定是一个字符，我们发现这样的情况下BC=C~B(~B是B的反转)这样就可以尝试将B并入下一个字符串，到最后如果没有下一个字符，B就成了多余后缀的一部分。
>
> 例如：|aabab|a|b|
>
> 可以尝试变成|a|ababa|b|
>
> 此时发现ababa不能合并，就可以把后面的baba并入最后一个|b|变成|babab|
>
> 此时仍不符合条件，就把baba向后推变成|a|a|b|abab，此时就变成最短划分了
>
> 此时第一个字符对应的一定子串是最短的，就可以重复上面操作使剩下的划分变成最短。

于是任何一种结果t都可以唯一表示成一种s的前缀的划分

我们现在知道了每一种结果串t都可以通过上面的方法唯一对应一个s的前缀的划分。现在再说明当这个多余的后缀的权值为0时，能表示成s的前缀的划分的串一定能表示成s的划分。

> 我们知道t可以由s的一段前缀生成，如果要满足条件1，那么剩余部分的权值必须为0
>
> 考虑将这段后缀与最后一段划分合并，不能合并当且仅当串是类似`abababababa...`的形式，这样最后一个划分一定是一个字符，可以将这个字符留在最后，将剩余的部分尝试跟前面的划分合并。
>
> 如果到第一个字符仍不能合并，那么这个串s一定是形如`abababa...`的，此时可以特判掉。

由于我们可以将所有串t唯一表示成s的前缀的划分，计数可以通过一个简单dp实现，记f_i表示有多少的串的最短划分是结尾为i的前缀，转移就是枚举下一个字符是a还是b，然后把权值等于s串的权值的前缀的dp值求和就是答案。注意特判掉没有相邻两个字符是相同的的情况。