Solution

Disposrestfully

August 3, 2019

1 无聊的日常

1.1 写在前面

出题人很良心地告诉了你,这套题不需要数据结构.

(某种意义上削弱了思维难度)

如果你并没有那么精于套路,A不掉$T2$的话,这可能就是你的签到题了.

1.2 一句话题意

N个点的树,边权均为1,求任意一个点x满足M条限制.

第i条限制为"x到节点A_i的距离加上x到节点B_i的距离不超过D_i".

1.3 正解

首先我们选择1号点为根dfs,求出每个点的深度.

考虑一条限制(A_i, B_i, D_i),令$W_i = max(0, \frac{dep[A_i]+dep[B_i]-D_i}{2})$.

可以发现$dep[x] \geq W_i$,也就是说每一条限制会给$dep[x]$一个下界.

我们找到最紧的下界,以及这个下界对应的限制.

在所有满足这组限制的点里面找一个深度最小的,如果这个点满足所有限制,那么它就是答案,否则无解.

四次dfs即可,正确性显然.

2 真正的巨佬

2.1 写在前面

或许是思维难度最低代码难度最高的一题.

数据水,解法套路,业务熟练的话直接就能切.

2.2 一句话题意

一句话讲不清.

2.3 ???

很明显这个问题可以被分成两部分.

要完成后续的计数工作,我们需要知道每座城市里有多少个$Fake$的人.

具体划分一下就是

$Part.1$ 算出每座城市里有多少人会$Fake$.

$Part.2$ 计数

2.4 Part.1

首先我们考虑两座城市u,v以及一条从u到v的边.

假设u中编号为i的人是$Fake$的,那么v中所有编号为j,使得$i \equiv j \ (mod\, gcd(S_u, S_v))$的人都会变得$Fake$.

再考虑从u到v的一条路径,路径上经过的点的S值的gcd为k,那么v中所有编号为j,使得$i \equiv j \ (mod\ k)$的人都会变得$Fake$.

那么对于一个强联通分量,所有点S值的gcd为k,如果其中存在一个编号为i的人是$Fake$的,那么这里面所有编号为j,使得$i \equiv j \ (mod\ k)$的人都会变得$Fake$.

如果我们知道了这个强连通分量里最后会有x个人变得$Fake$,那么对于一个属于该强连通分量的点u,其中应该有$x \times \frac{S_u}{k}$个人变得$Fake$.

现在我们只需要考虑缩点之后的图.

因为原图是竞赛图,所以缩点之后的新图存在一条哈密顿路径,在这条哈密顿路径上点的编号是递减的,每个点只会对它之后的点产生影响,可以暴力转移.

然后我们就能求出每个城市里有多少人会$Fake$了.

2.5 Part.2

现在我们把问题转化成了这样:

N个变量,其中第i个变量的取值可以是$[Mn_i, Mx_i]$中的任一整数,从前A大的变量中选出B个,求可能选出的集合的数量.

我们枚举一个变量u,钦定它作为被选中的B个中最小的(有多个取编号最大的).

计算出满足$Mn_i > Mx_u$的变量i的数量Cnt_1,以及满足$Mx_i \ge Mx_u \ge Mn_i$的变量的数量Cnt_2,枚举我们在这Cnt_2个变量里我们选了x个,答案加上$C_{Cnt_2}^{x} \times C_{Cnt_1}^{B-1-x}$,注意一下枚举边界即可.

一些具体的实现细节可以参考标程.

3 隐蔽的居所

3.1 写在前面

这题应该是有两个做法.

正解做法不是那么的套路,不过也没有那么难想到.

另外的那个做法更好想,就是比较(可能是非常)难写.

如果你不会做另外两道题的话就能拿那个做法来签到.

3.2 一句话题意

N个点,每个点和编号差不超过K的点连边.

从这些边中擦去M条,求哈密顿回路的数量.

3.3 Sol.1

$K \le 2$时很好处理,我们现在只考虑$K = 3$的情况.

考虑把数字从小到大放到环上去,显然数字i的放法只和$i-1$,$i-2$和$i-3$的位置有关.于是我们有了一个非常显然的Dp做法.

设$dp[i][0/1][S_1][S_2]$表示当前放到了数字i,$i-1$,$i-2$和$i-3$的相对顺序是顺时针/逆时针,这三个数是否放在端点的状态为S_1,这三个数是否相邻的状态为S_2.

人有多大胆,地有多大产.

3.4 Sol.2

上面那个做法怎么看怎么不优秀,我们需要一个更好的做法.

首先把编号i变成编号$N-i$.

那么题目需要我们求的,是一个以0开头,以1/2/3结尾,并且相邻两项之间的差不超过3,某些数字不能放在另一些数字右边的数列.

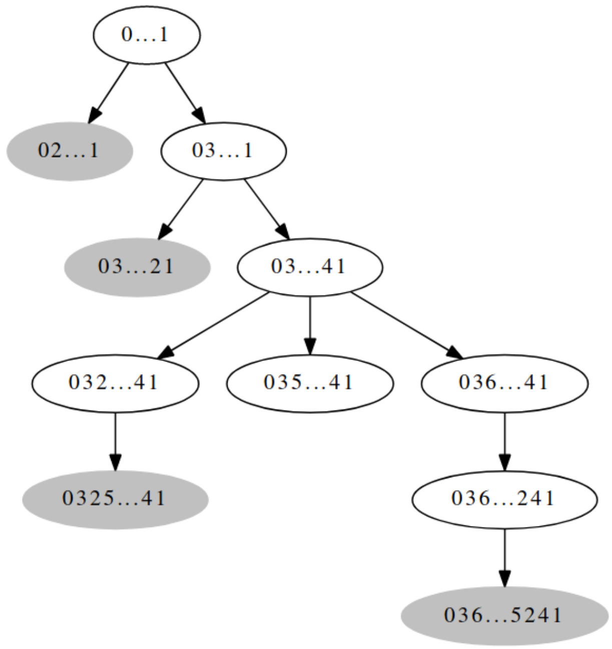
考虑换一种方式dp.

设$f[i]$表示以i开头以$i+1$结尾,所有元素都在$[i,N)$中且符合题意的数列的数量.

$g[i]$表示以$i+1$开头以i结尾,所有元素都在$[i,N)$中且符合题意的数列的数量.

很显然在小范围内,我们可以通过搜索来求出这两个值.

那么我们现在来看一看这个搜索会搜出个啥玩意.



诶我们发现$f[i] = g[i+1] + g[i+2] + g[i+4] + g[i+5]$.
而且$f[i]$和$g[i]$的状态是对称的,那么$g[i] = f[i+1] + f[i+2] + f[i+4] + f[i+5]$.
这个dp只适用于$i \le n-8$的情况,但问题不大,对于i更大的情况可以真的去搜.
那么现在的问题就是,如何统计分别以1/2/3结尾的数列的数量.
很显然$Ans_1 = f[0]$,然后我们手玩一下(搜也行).

$$01 \diamond \ldots \diamond 2 \quad 03 \diamond \ldots \diamond 412 \quad 0314 \diamond \ldots \diamond 52.$$

$$012 \diamond \ldots \diamond 3 \quad 014 \diamond \ldots \diamond 523 \quad 01425 \diamond \ldots \diamond 63 \quad 0214 \diamond \ldots \diamond 3 \quad 025 \diamond \ldots \diamond 413.$$

那么$Ans_2 = f[1] + f[3] + f[4]$,$Ans_3 = f[2] + f[4] + f[5] + g[3] + g[4]$.
在dp的时候考虑一下附加的限制条件就行了.
具体实现可以参考标程.

4 附加题

第一想法显然是二分答案,模拟判定.
然后再仔细一想,这东西并没有单调性.
然后再仔细一想,相邻两个可行解之间的距离不超过2000.
二分一个界出来,然后往下面枚举即可.