# Prob2 Solution

2018 年 11 月 4 日



## 1 road

这道题，我们可以考虑以 s 为根，那么问题转化为给定一条路径 a b，最优的选择一个点 c，使得将 s 到 c 的边变为 0，然后 a 到 b 的边权和最小。

那么，我们可以先求出 a 与 b 的 lca，然后再去考虑我们要覆盖的部分是路径 a 到 lca 上的还是 b 到 lca 上的。假如求 a 到 lca 上的，我们要求的是找到一个 c 使得 a 到 c 的边权和最小。

因此，我们只需要通过倍增，维护一个 lca[i][j] 表示从 j 往上跳 $2^i$ 到达的节点，ssum[i][j] 表示从 j 到 lca[i][j] 的路径上边权和，以及 minn[i][j] 表示从 j 到 lca[i][j] 路径上，选择最优的 c 所得到的最小化的从 j 到 c 的路径上边权和。

## 2 bride

### 2.1 整体思路

先枚举所有贿赂的方案，在此基础上枚举所有可能的投票方案。

### 2.2 实现

第一层枚举可以直接 DFS，这里要注意避免重复枚举方案。若每次枚举贿赂给 $n$ 个人中的一个，则其复杂度为 $O(n^k)$；若每次枚举的范围从上一个刚贿赂过的人开始，则复杂度为 $O(C(n+k-1,n-1))$。复杂度的差异源自方案的重复枚举次数。

第二层枚举可以直接 DFS, 依次枚举某个人投了支持还是反对的票, 递归的时候记录当前投票结果发生的可能性、支持的议员和反对的议员的数量差、投了支持的议员的威望值之和。复杂度为 $O(2^n)$。

综上，复杂度：$O(C(n+k-1,n-1)\times 2^n)$

# 3 climb

每次从某个点到根染色，每次询问某个点被打了多少种标记。



## 3.1 算法一

开一个数组 $a[i,j]$ 表示第 $i$ 个点的第 $j$ 种标记是否被打过，每次询问可以 $O(total)$ 回答，其中 $total$ 是标记数。

或使用一个计数器，每次有某个 $a[i,j]$ 从 False 变 True 则 $i$ 的计数器加 1，询问可以做到 $O(1)$。

每次修改暴力打标记。

明显的，如果 $max(G_i)$ 大于 $M$，则可以通过离散把数量级下降到 $M$。

时间复杂度 $O(NM)$，空间复杂度 $O(Nmax(G_i))$

### 3.1.1 优化

如果 $a[i,j]$ 已经是 True，那么已经可以结束打标记了，因为 $i$ 的所有祖先一定都已经打过了标记 $j$。

时间复杂度 $O(Nmax(G_i))$，空间复杂度 $O(Nmax(G_i))$

## 3.2 算法二

由于每次修改影响到的仅是修改点的祖先，可以换一种储存方式，直接把标记打到点上。每次询问一个点只需要查询该点为根的子树下有多少不同的标记。

这样可以使用 DFS 序，对每种标记开一个树状数组，对每个标记，查询该子树下是否有标记

时间复杂度 $O(Mlog(N)+Nmax(G_i)log(N))$,空间复杂度 $O(Nmax(G_i))$

### 3.3 算法三

发现算法二没有必要使用树状数组维护，只需要每种标记分开处理，处理只需要简单地遍历整棵树

时间复杂度 $O(Nmax(G_i))$，空间复杂度 $O(N)$

### 3.4 算法四

首先我们仅考虑只有一种标记的情况，如果仅有一种标记，那么我们可以把被打标记的点进行 $+1$ 操作，那么如果最后，一个点的子树和大于 $1$，该点就是有标记的。

但是这样处理不了多种标记的数量，所以要把多余的 $+1$ 标记减去，而多余的 $+1$ 标记的形成是两个操作拥有公共祖先，所以在两个操作点的 LCA 处打上一个 $-1$ 标记就可以避免多加的情况，原来的两个操作就变成了一个操作了，可以证明 $-1$ 标记只会打 $O(M)$ 个。

如此就可以拓展到多种标记，而标记数就是该点的子树权值和。为了方便打 $-1$ 标记，我们可以交换操作顺序，把同一种标记的一起做，而 $-1$ 标记所打的点就是这些相同标记的操作点按 DFS 序排序之后相邻两点的 LCA。

时间复杂度为 $O(MlogN + N + MlogM)$