A:

对于 30%的数据，合理的暴力即可。

我其实出了一部分的数据答案是-1，骗分的同学快感谢我。

对于 100%的数据：

先处理一个子问题：对于任何一条边，去掉该边后端点 i 到 T 的最短路径 f[i]。

从 T 跑出一棵最短路树，因为如果边不在最短路树上，那么依然是最短路长度，否则的话，考虑将树上 y 到 fa[y]该边去掉，则是在子树中取一个点，跳横跳边再往上到根，也就是 dist[x]-dist[y]+edge[x,p]+dist[p]。

我们可以先对于每个点求出最小的 dist[x]+edge[x,p]+dist[p]，考虑非树边(x,p)，若 x 在 z 的子树内，p 不在 z 的子树内，则该值就可以对 z 的 f 产生贡献。我们枚举每一条非树边，对于(x,y)，则将 x 到 lca(x,y)之下的每个点都更新掉，用树链剖分实现。然后把所有的 f[i]都减去 dist[i]就得到真正的 f[i]。

得到 f 之后，从 T 向 S 跑最短路，但是更新答案的时候要用 max(d[x]+edge[p].w,f[edge[p].adj])来更新答案。

意义为如果走最短路比我按按钮结束要劣，那我就等着。

这一步也可以用二分答案也是另一个求解方法。

10%的点是怕大家写萎了送给大家的中档数据，实际上可以爆枚删去所有边求出这个数组，我好怕我数据造得不够强...

Never mind，A 题温暖送给大家。

《见面会》题解

为方便起见，我们可以统计每场见面会的人数的平方和，最后减去总人数再除以二就是答案。

首先对每一个编号为i的员工，预处理出最小的j，使得第$j+1$号员工到第i号员工之间有公共时间。然后就是一个dp：

$$dp[i] = \min_{j \le k < i} (dp[k] + (i-k)^2)$$

直接转移就可以得到$O(n^2)$的做法。

考虑使用斜率优化：

$$dp[i] = \min_{j \le k < i} (dp[k] + k^2 - 2ki) + i^2$$

可以使用线段树维护凸壳，dp时查询下标位于$[j,i)$，$x=i$处的最小值。dp后插入$y=-2kx+k^2+dp[k]$这段斜线即可。总复杂度为$O(n \log n)$

Bonus: 在线性时间内解决该问题。

C

首先显然等他们都走到环上就行。那之后答案不变。
考虑修改操作。
所有操作肯定连成一个联通块，讨论这个联通块上的环是人造的还是天然的。
人造环可以发现一定是自环，且其他连接上的联通块位置无关，只需要是拆环连上去就行。
考虑天然环，枚举这个天然环是谁，其他联通块修改一条边连上去。
0：只有大小为 1 或 2 的环，二分染色即可知道连接到二元环的贡献，随后直接贪心即可。
1：是一个排列，只有环。显然若核心是自然环，一定取最小环，一定将最大的环连接上去。
2，3，4：
考虑连接自然环的拆环。发现一定拆环边而不是树边。
连接上去的贡献只需要取环上最多人的点即可。
到这里直接暴力自然环，复杂度已经是 O（N^2）。
5，6，7，8，9：
考虑每一次拆开下一条环边时贡献的变化。发现就是把原本的环头带着连接的树节点放到环尾。
所以每次修改可以 O(子树点数)。每个联通块对天然环的最大贡献可以 O(联通块点数)计算得到。
联通块的选择可以排个序贪心。(可以计数排序)所以每次枚举自然环后乘上的复杂度是 O(n)。
发现天然环具体是谁对于联通块贡献效果是无关的，只与天然环长度有关。而长度最多只有 O(sqrt(n))种。
因此枚举可能的环长，O(n)求出所有联通块的贡献，把贡献最大的的环长等于目前枚举值的拿走作为自然环，剩下的贪心即可。
总复杂度 O(n^1.5)