T1 BZOJ 4034

数据结构裸题

一种非常显然的方法是直接树剖，复杂度 $O(n\lg^2 n)$， 链加就跳重链，子树加就直接在子树对应的 DFS 序上修改一下就行了。

另外一种更优秀的方法基于题目的一个很好的性质，每次修改都是从根懂啊一个点的路径，因此我们可以利用入栈出栈来解决。

T2 BZOJ 1774

这题看起来和最短路差不多，但是却多了一个限制，要加上路径上的点权的最大值。

考虑 $floyd$

对于这类题目，我们有一个非常显然的做法是枚举最大值。我们把点按照他们的权值从小到大排序

然后按照这个顺序来枚举中间点 k 来更新

我们用 $\mathtt{dist}(i,j)$ 表示 i 到 j 之间的距离， $\mathtt{ans}(i,j)$ 表示 i 到 j 之间的答案。

考虑暴力转移时需要枚举点来找到点权最大的点，如何优化这个枚举?

我们可以将点按点权从小往大排序，枚举 k，i，j从 1 到 n 实质上枚举的是排序后的编号，这么做的好处是转移时点权最大的点一定在k， i ， j三点之中，这样就避免了枚举点这一操作。

注意如果要这样处理一定要让 `dist` 和 `ans` 一起更新，否则会出现走了路径 A , 加了路径 B 的最大值的问题。

T3 BZOJ1690

典型的分数规划问题

首先我们可以知道，这个题虽然可以选择多个环，但是答案一定是只选择一个环的情况。

假设存在多个环，我们发现其中任意一个都可以单选。

假想有两个环，一个比值大，另一个比值小。他们合起来的比值肯定小于其中大的哪一个环。加上走重点或边代价变大，收益不变的情况，显然答案对应的路径肯定能只用一个环表示。

接下来就是分数规划的常规操作：

我们要最大化$\frac{\sum_{i=1}^{n}v_i}{\sum_{i=1}^{n}e_i}$ （n是这个环的点数）

我们不妨让这个答案是 ans, 注意 ans 是一个最大值，所以对于任意一个环都会有

$$\frac{\sum_{i=1}^{n}v_i}{\sum_{i=1}^{n}e_i} \le ans \to \sum_{i=1}^{n} v_i \le ans \sum_{i=1}^{n} e_i = \sum_{i=1}^{n} ans \cdot e_i \to \sum_{i=1}^{n} v_i - ans \cdot e_i \le 0$$

我们考虑二分 ans, 设真实的到的答案是 ans, 二分得到的答案是 $ans1$,当二分得到的 $ans1 < ans$ 时,图中就会出现正权环；如果图中没有任意一个正权环，说明我们二分的 $ans1$ 已经足够大了，答案应该比 $ans1$ 小一些。

所以我们用 SPFA 来跑最长路，同时找正权环即可。