

Solution

Dez

2017 年 10 月 19 日

1 数列求和

要求一个数列的所有子段的元素积之和。发现每新增一个元素，形成的序列是它本身和所有结尾为它的上一个元素加上它本身。所以每次记下上一个元素的贡献，这个元素的贡献也能求出来，最后相加即可。时间复杂度 $O(n)$ 。

2 路径统计

观察题意发现，图中只可能存在树和环。树上边的贡献统计很简单，等于这条边前面能经过的点与后面能到达的点的乘积再乘上自己的权值。而环上的点比较麻烦，既要考虑环中的路径，又要考虑它外接的树的路径。考虑每次把环拆成链来考虑，这样是可行的因为你不可能在环上转圈，经过的点肯定不会再走了。唯一的区别在于你在考虑环上的每个点的时候，它链的长度都是环的大小（链显然是不一样的）。假设存在一个大小为 n 的环，我们先选择一个起点，可以发现起点连出的边会被统计 $n-1$ 次，下个点的边则是 $n-2$ 次，下下个则是 $n-3$ 。依次类推，发现它的贡献次数是递减的，那么当我们把起点换成下一个点时，后面所有边的贡献次数都加一，而原来的第一条边次数变为 0。我们有一个很方便的办法转移，记下环上路径的权值和，每次转移起点时加一遍路径和，再减去 n 倍（不是 $n-1$ ）原来的起点边，就得到新的贡献。最后对于枚举的每个点的贡献，还要乘上它外接的树的大小。对于不能到达的点对，我们可以记下能到达的点对的数量，而整个图有 $n \times (n-1)$ 个点对。拓扑排序是必需的，又因为统计树时不能带环，所以我们要缩点。使用 *Tarjan* 求强联通分量算法，时间复杂度 $O(n)$ 。

3 小店购物

因为所有物品有无限个，所以每次选择一个物品至花费少会超过 $\frac{k}{2}$ 。所以我们最多只需要选择 $\log k$ 次物品。我们考虑用权值线段树维护价格区间内价值的最大值。但是又因为每个价值上可能有多个物品，物品有可能被删除，所以我们对于每个价格都要维护它的物品集合。这里使用离散化加权值线段树加 *std::multiset*，时间复杂度 $O(n \log^2 n)$ 。