

小龙的旅行

难度: 提高-

s 为233333, 顶点最多为 10^5 , 因此实际上就是求一个字典序最小的序列.

由于多次经过已经经过的点是不算的, 因此使用一个优先队列维护尚未遍历, 但是马上可以遍历的点的集合, 每次从中选最小的编号点即可.

小龙的平均值

难度: 提高

显然先二分区间平均值 m , 然后快速计算有多少个区间平均值大于等于 m .

对于一个区间平均值, 数学表达为:

$$a[l] + a[l+1] + \dots + a[r] \geq (r-l+1) * m$$

即

$$(a[l] - m) + (a[l+1] - m) + \dots + (a[r] - m) \geq 0$$

令 $b[i] = a[i] - m$, sum 为 b 的前缀和, 则公式变为

$$sum[r] - sum[l-1] \geq 0$$

即

$$sum[l-1] \leq sum[r]$$

暴力枚举 r 端点, 使用线段树/树状数组维护角标为 $sum[i]$ 的出现次数, 进行区间求和即可

需要对数字进行离散化

小龙的花费

难度: 提高+

对于没有专利费限制的部分, 设置 $f[i][2]$ 代表走到第 i 列, 处于第一行或者第二行时, 花费的最小花费, 枚举每次决策进行转移即可.

对于有专利费限制的部分:

前70分(暴力分较多, 对尚未学到网络流知识的同学留了70分的可做分), 均可以通过状压暴力来完成. 对于 $|i-j| \leq 12$ 的数据, 可以维护当前节点往前的12个节点的行走情况, 每次计算下 m 个限制是否需要加上. 可以通过把限制根据后面的节点放在对应的角标上进行优化.

对于全分, 需要用到最小割的经典模型.

核心思路为: 新增一些边, 让一上一下or一下一上的边无法直接割掉后完成切断 S 到 T , 导致必须带上额外花费, 以此强制补上了上下行的花费代价.

设第一行的代价为 $a(i, i+1)$, 第二行代价为 $b(i, i+1)$

建模，凑最小割：

- 设点 i 代表通过第一行从 i 走向 $i + 1$, S 向点 i 连流量为 $a(i, i + 1)$ 的边. 对于两头的边需要根据题意特殊表示下
- 设点 i' 代表通过第二行从 i 走向 $i + 1$, 先将 i 向点 i' 连流量为 $b(i, i + 1)$ 的边, 然后 i' 向 T 连流量 INF
- 对于 i 列上下行走的代价, 建立 $i - 1$ 指向 i' , 以及 i 指向 $(i - 1)'$ 的边, 流量为对应的代价
- 对于专利限制, 则和第三项类似, 建立 j 到 i' 的边, 流量为对应的代价

这样跑出来的最小割, 刚好满足题意

小龙的苹果树

难度: 省选

对于 20% 的数据

我们枚举两个点求解其树上距离即可。

对于 40% 的数据

我们考虑 dp, 设 $w_x = dep_1(x) + dep_2(x)$, 那么所求等价于 $\max \{w_x + w_y - 2 \text{dep}(LCA(x, y))\}$, 枚举 LCA, 只需要分别统计其不同子树内 w_x 的最大值与次大值即可。复杂度 $\mathcal{O}(n)$

对于测试点 5,6

保证修改随机生成, 同时 T_2 是一张菊花图。这意味着我们可以期望单次修改只会影响单个点的 dep 值。于是新的答案只有两种:

- 以原来的点对 (u, v) 为答案。
- 以 $(x, ?)$ 为答案, x 为本次修改的点。

于是我们只需要统计 $\max \{w_x + w_y - 2 \text{dep}(LCA(x, y))\}$ 即可, 注意到 LCA 只能是 x 的父亲上的节点, 我们只需要维护 $\max \{w_y - 2 \text{dep}(LCA)\}$, 可以通过树剖维护非重链信息处理

对于全分

我们对于原图的每个节点 i 建立一个虚点 i' 连接 i , 边权为 $dep_2(i)$, 那么原题所求等价于新树上的直径。

树的直径可以 dp 处理, 另一种方法是找到离一个点最远的点 u , 然后考虑离 u 最远的点 v , 那么 $u \rightarrow v$ 即为树的直径。

- 可以证明离点 x 最远的点一定是直径中的一个点 u 。
- 容易发现离 u (直径上的一点) 最远的点 v 必然为直径上的另一点。

我们考虑如何单点修改 (即仅增大某一个点的 dis)。

注意到每次只增大边权, 我们将单次增加视为在 i' 下面挂一个新的节点 i'' , 边权为 c

我们发现这个模型是在一棵树上加入一个节点的形式, 那么可以证明新的直径要么是原来的直径, 要么是原来的直径中的一个点与新加入的节点构成的。

所以我们维护原直径的两个节点，每次加入新节点的时候维护直径即可，这里只需要查询节点到两个点的距离即可，涉及到查询LCA,.

由于 T_2 为随机生成的, 我们单次修改涉及节点数期望为 $\log n$, 所以暴力单点修改的复杂度为 $\mathcal{O}(m \log^2 n)$