USACO趣题选讲

T1 题目描述

Bessie 正在参加远足旅行！她当前正在旅行的路线由编号为 $1 \ldots N$（$1 \leq N \leq 10^5$）的 N 个检查点组成。

有 K（$1 \leq K \leq 10^5$）张票可供购买。第 i 张票可以在检查站 c_i（$1 \leq c_i \leq N$）以 p_i（$1 \leq p_i \leq 10^9$）的价格购得，并且可以用其进入所有检查站 $[a_i, b_i]$（$1 \leq a_i \leq b_i \leq N$）。在进入任何检查站之前，Bessie 必须已购买一张允许其进入该检查站的票。一旦 Bessie 可以前往某一检查站，她就可以在未来的任何时候回到该检查站。

对于每一个 $i \in [1, N]$，如果 Bessie 最初只能进入检查点 i，输出使得可以进入检查点 1 和 N 所需的最低总价。如果无法这样做，输出 -1。

P7984 [USACO21DEC] Tickets P

最短路好题啊。

第一眼看过去，发现好像直接线段树优化建图一下就好了。

但是问题是，某些路径会算重。

那么考虑一个点，到 1 和 n 距离和最小的情况是什么?

一定是开始可能走一段相同的，后来走一段不交的。

因为如果之后路径有交，那我干嘛闲的没事分开呢?

所以我们直接线段树优化建反向边，从 1 和 n 开始分别跑最短路。

然后再将每个点的起始权值设定为到 1 和 n 的距离之和，跑最短路即可。

复杂度 $O(n\log^2 n)$。

T2 题目描述

Bessie 有一个数 $x+0.5$，其中 x 是某个 0 到 N 之间的整数（$1 \leq N \leq 5000$）。Elsie 正试着猜这个数。她可以以如下形式对于某个 1 到 N 之间的整数提问：「i 是大了还是小了？」如果 i 大于 x，Bessie 会回答 "HI!"，如果 i 小于 x 则回答 "LO!"。

Elsie 想到了以下猜测 Bessie 的数的策略。在进行任何猜测之前，她创建了一个长为 N 的一个排列。然后她遍历这一排列，按排列中的数的顺序依次猜数。然而，Elsie 会跳过所有不必要的猜测。也就是说，如果 Elsie 将要猜某个数 i，而 Elsie 之前已经猜过了某个 $j < i$ 并且 Bessie 回答 "HI!"，Elsie 不会再猜 i，而是继续猜序列中的下一个数，反之亦然。可以证明，使用这一策略，对于 Elsie 创建的任意序列，她都可以唯一确定 x。

如果我们将所有 Bessie 回答的 "HI" 或 "LO" 拼接成一个字符串 S，那么 Bessie 说 "HILO" 的次数为 S 等于 "HILO" 的长为 4 的子串数量。

Bessie 知道 Elsie 将要使用这一策略，并且已经选定了值 x，但她不知道 Elsie 会使用什么排列。你的目标是对于所有 Elsie 可能选用的排列，计算 Bessie 说 "HILO" 的次数之和，对 10^9+7 取模。

P7986 [USACO21DEC] HILO P

考虑一种朴素的 dp，记 $dp_{l,r,i,j,0/1}$ 表示目前确定答案在区间 $[l,r]$ 内，问了 i 次，已有 j 个 “HILO”，上次是 HI 还是 LO，目前排列数。

枚举排列中下一个数转移，总复杂度 $O(n^5)$。

如何优化呢?

P7986 [USACO21DEC] HILO P

j 这一维是否有用?

考虑记录两个 dp 数组，一个表示总答案，一个表示方案数，用这种思想就省略掉了 j 这一维。

复杂度优化为 $O(n^4)$。

P7986 [USACO21DEC] HILO P

考虑排列数量，等价于每次随机选一个未选过的算概率，于是可以省略掉 i 这一维。

于是我们状态变成了 $dp_{l,r,0/1}$ 表示确定 x 在 $[l,r]$ 内，当前是 HI/LO，出现这种情况的概率，然后枚举一个 $l<k<r$ 转移。

复杂度优化为 $O(n^3)$。

P7986 [USACO21DEC] HILO P

然后发现这个东西可以对 l, r 两维分别记两个前缀和。

大概就是 $suml_{l,r} = \sum_{i=0}^{l} dp_{i,r}$ 和 $sumr_{l,r} = \sum_{i=r}^{n} dp_{l,i}$，然后转移即可。

复杂度 $O(n^2)$。

注意一个问题就是这题空间限制 256MB，只够开两个 n^2 的数组，怎么办？

P7986 [USACO21DEC] HILO P

记录总答案数的 dp 其实是不需要的。

直接在方案数的基础上连续做两步转移算系数贡献到答案就好。

然后转移的 dp 数组本身我也不用求出来，毕竟转移当中有用的只有前缀和和。

所以就只用开 $suml, sumr$ 两个数组就够了。

T3 题目描述

Farmer John 的 N 头奶牛（$2 \le N \le 3 \cdot 10^5$），编号为 $1 \dots N$，排列成 $1 \dots N$ 的一个排列 $p_1, p_2, \dots, p_N$。另外给定一个长为 $N-1$ 的字符串，由字母 U 和 D 组成。请求出最大的 $K \le N-1$，使得存在 p 的一个子序列 $a_0, a_1, \dots, a_K$，满足对于所有 $1 \le j \le K$，当字符串中第 j 个字母是 U 时 $a_{j-1} < a_j$，当字符串中的第 j 个字母是 D 时 $a_{j-1} > a_j$。

P8277 [USACO22OPEN] Up Down Subsequence P

朴素的想法是设 $f_{i,j}$ 表示排列的前 i 个当中已经选择了 j 个位置进入子序列，且最后一个选择进去的方案数，然后枚举转移，复杂度 $O(n^3)$。

如何优化?

P8277 [USACO22OPEN] Up Down Subsequence P

考虑对于一个位置 i，是不是只有最长的子序列有用?

好像很有道理的样子。

我们先假定这是对的。

那么转移的时候，如果求 U 的答案，就查询在它前面，且值小于它的那些位置的答案。

- 如果后面一个就是 U，答案取这个 U。
- 如果后面一个不是 U，答案取前面（含本身）第一个 U。

丢到权值线段树或者树状数组上查一下前后缀 max 即可转移。

复杂度 $O(n\log n)$。

P8277 [USACO22OPEN] Up Down Subsequence P

但是，比较长的序列满足要求，短的并不一定会满足！

为什么这是对的？

考虑转移的过程，以 U 为例，“如果后面一个就是 U，答案取这个 U”这种情况显然不会算大或算小答案。

以下我们只需考虑“如果后面一个不是 U，答案取前一个 U”这种情况。

P8277 [USACO22OPEN] Up Down Subsequence P

这样显然答案一定不会算小，因为它再怎么也碰不到下一个 U。

答案想想可能会算大，但是一定不会更新全局最优解，取出转移点处的合法情况，设这个 U 代表 $a_x < a_y$，分讨两种情况：

- $a_x < a_i$，可以把最后一个 U 改成 $a_x < a_i$ 是合法情况。
- $a_x > a_i$，那么一定有 $a_y > a_i$，那么在接下来的转移中：
    i. 如果遇到一个 U，那么一定不会占到优势。
    ii. 如果遇到一个 D，那么一定不会比这个转移点更优。

所以这是对的。

T4 题目描述

农民约翰有 $N(2 \le N \le 2 \cdot 10^5)$ 台拖拉机, 其中第 i 台拖拉机只能在序列 $[l_i, r_i]$ 内使用。拖拉机有左端点 $l_1 < l_2 < \cdots < l_N$ 和右端点 $r_1 < r_2 < \cdots < r_N$. 有一些拖拉机是特别的。

如果 $[l_i, r_i]$ 和 $[l_j, r_j]$ 相交，则两台拖拉机 i 和 j 是相邻的。 约翰可以从一辆拖拉机转移到任何相邻的拖拉机上。两台拖拉机 a 和 b 之间的路径由一个传输序列组成，这样序列中的第一个拖拉机是 a，序列中的最后一个拖拉机是 b，并且序列中的每两个连续的拖拉机相邻。 保证拖拉机 1 和 拖拉机 N 之间有一条路径。路径的长度是转移的数量 (或等价地，其中拖拉机的数量减去 1)。

给定 $Q(1 \le Q \le 2 \cdot 10^5)$ 组询问，每次给定 a 和 $b(1 \le a < b \le N)$。 对于每组询问，你需要回答两个问题：

1. a 到 b 的最短路径。
2. 在传送次数的最少的情况下，有多少个特殊拖拉机的区间可能被某条最短路经过。

P9019 [USACO23JAN] Tractor Paths P

首先考虑第一问怎么求最短路?

显然为了保证最小，一个区间一定会转移到与它有交的最靠右的区间。

于是直接二分出转移点，然后倍增计算最短路即可。

P9019 [USACO23JAN] Tractor Paths P

要求所有最短路可能经过的点，怎么办？

从一个点 x 出发到 $y(x<y)$，最短路的长短是随 y 的增加而增加的。

考虑最后一步是怎么走的。

最后一步，只要是能一步到达终点的，都可以作为倒数第二个到达的点。

这样说，最后一步可以从之出发的点一定是一个区间。

往前推，所有能够到达最后一步可行点也是一个区间。

以此类推，每一步的可行点都是一个区间。

我们要求所有区间的长度和，可以转化为所有右端点编号减去所有左端点编号，倍增时候顺便维护一下即可，复杂度 $O(n\log n)$。

T5 题目描述

给定一个包含 N 个结点和 M 条边的有向图（$2 \leq N \leq 10^5, 1 \leq M \leq 2 \cdot 10^5$），Farmer John 的奶牛们喜欢玩以下的双人游戏。

在图中的不同结点上放置两个指示物。每一回合，一名玩家，脑，选择一个需要沿某一条出边移动的指示物。另一名玩家，蹄，选择沿着哪条出边移动该指示物。两个指示物在任何时刻不允许处于同一个结点上。如果在某些时刻蹄不能做出合法的行动，则脑获胜。如果游戏可以无限进行下去，则蹄获胜。

给定 Q 个询问（$1 \leq Q \leq 10^5$），包含两个指示物所在的初始结点。对于每个询问，输出哪名玩家获胜。

P8276 [USACO22OPEN] Hoof and Brain P

hint：缩点，真的有用吗？

P8276 [USACO22OPEN] Hoof and Brain P

发现一件事情，就是只有 1 个出度的点，没有存在价值。

首先它不表示任何选择，规定了只能这样走。

其次如果能在这个点发生堵截现象，那么在它指向的点也可以。

所以这启发我们缩 1 度点。

P8276 [USACO22OPEN] Hoof and Brain P

首先先缩掉零出度的点，毕竟这些点是一定会走上不归路的。

然后对于一度点，缩掉的过程本质上就是把所有连向它的点连向它的出点，再除去重边。

发现这一过程暴力做不行，于是考虑用 set 维护入边，并查集合并点。

我们缩掉 $x \to y$ 这条边的时候，暴力更新所有出边肯定复杂度爆了。

于是考虑启发式合并。（其实是 NOI2024 day2T3 做法的第二部分）

对于入边集合，小的合并进大的当中。

至于用 set，是因为要在过程中去除重边，更新出度，一直做。

P8276 [USACO22OPEN] Hoof and Brain P

最后一定会缩成一个含有自环的图，自环虽然可能出度为 1，但我们并不缩起来。

然后判定的话，如果两个点中存在被当成零度点删掉的，直接脑获胜。

如果这两个点最后会到一个自环里面，也是脑获胜。

于是判断一下缩完之后初始情况下两个点所位于新图的点是否相同。

- 若相同，则一定有办法把一个卡住，脑获胜；
- 若不同，则蹄获胜。

总复杂度 $O(n\log^2 n)$。

T6 题目描述

Farmer John 的农场可以用一个带权有向图表示，道路（边）连接不同的结点，每条边的权值是通过道路所需的时间。每天，Bessie 喜欢从牛棚（位于结点 1）经过恰好 K 条道路前往草地（位于结点 N），并希望在此限制下尽快到达草地。然而，在某些时候，道路停止维护，一条一条地开始破损，变得无法通行。帮助 Bessie 求出每一时刻从牛棚到草地的最短路径！

形式化地说，我们从一个 N 个结点（$1 \le N \le 300$）和 N^2 条边的带权有向完全图开始：对于 $1 \le i, j \le N$ 的每一对 (i, j) 存在一条边（注意存在 N 个自环）。每次移除一条边后，输出从 1 到 N 的所有路径中经过恰好 K 条边（不一定各不相同）的路径的最小权值（$2 \le K \le 8$）。注意在第 i 次移除后，该图还剩下 $N^2 - i$ 条边。

路径的权值定义为路径上所有边的权值之和。注意一条路径可以包含同一条边多次或同一个结点多次，包括结点 1 和 N。

P8906 [USACO22DEC] Breakdown P

hint：$K \leq 8$ 暗藏玄机，而且只需要求 $1 \sim n$ 的最短路。

于是只需要解决这样的问题：从 1 出发经过 4 条边走到所有点的最短路。

对 1 做和对 n 做同理，每次能 $O(n)$ 求出答案。

P8906 [USACO22DEC] Breakdown P

考虑对于 $1 \to i \to j \to k \to l$，如果更改了 $x \to y$ 的边权，会影响什么。

我们维护 $disk_{x,y}$ 表示 x 到 y 走 k 条边的最短路。

首先，加一条边 $dis2_{x,y}$ 发生改变的对只有 $O(n)$ 个。

如果边处在 $j \to k$ 或 $k \to l$，那说明 $j \to l$ 发生改变，而改变一条边边权影响对于 $a \to b \to c$ 这样路径的数量是是 $O(n)$ 级别的，我们枚举 j 更新即可。

如果边处在 $i \to j$，那说明 $1 \to j$ 发生改变，而这种改变最多只有 1 个，那我们枚举 l 更新还是 $O(n)$ 的。

如果边在 $1 \to i$ 怎么办？思考半天好像只能分析出枚举 j, l 更新的 $O(n^2)$ 做法。

但是好像又没关系？从 1 出发的边也就 $O(n)$ 条，总复杂度还是 $O(n^3)$ 没变。

总复杂度 $O(n^3)$ ，真妙的题目！

谢谢大家